

M A R C D A V I S

www.marcdavis.me

P U B L I C A T I O N S

info@marcdavis.me

Media Streams: Representing Video for Retrieval and Repurposing

Bibliographic Reference:

Marc Davis. "Media Streams: Representing Video for Retrieval and Repurposing." Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1995. Readers: Professor Kenneth Haase, Professor Seymour Papert, Professor Henry Jenkins.

Media Streams: Representing Video for Retrieval and Repurposing

Marc Eliot Davis

B.A., Wesleyan University (1984)
M.A., University of Konstanz (1987)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 1995

© Massachusetts Institute of Technology, 1995. All rights reserved.

Author
Program in Media Arts and Sciences
January 13, 1995

Certified by
Kenneth B. Haase
Assistant Professor
Program in Media Arts and Sciences, MIT
Thesis Supervisor

Accepted by
Stephen A. Benton
Chairman, Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Media Streams: Representing Video for Retrieval and Repurposing

Marc Eliot Davis

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning,
on January 13, 1995, in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract

Current computing systems are just beginning to enable the computational manipulation of digital video. Because of the relative opacity of video, it must be represented in order to be manipulable according to its content. Knowledge representation techniques have been implicitly designed for representing the physical world and its textual representations. Video poses unique problems and opportunities for knowledge representation which challenge many of its assumptions about the structure and function of what is represented. The semantics and syntax of video require representational designs which employ fundamentally different concepts of space, time, character, action, identity, and transition. In particular, the effect of the syntax of video sequences on the semantics of video shots requires that representation and retrieval technologies clearly articulate the differences between the sequence-dependent and sequence-independent semantics of video data.

Implementing these ideas, *Media Streams* uses a stream-based, semantic, memory-based representation with an iconic visual language interface of hierarchically structured, composable, and searchable primitives to annotate video for content-based retrieval. *Media Streams* addresses problems of annotation convergence and human-system communication by creating a standardized language for representing video content in a global media archive. The system introduces new retrieval-by-composition methods which reinvent video editing as a process of logging and retrieval. *Media Streams* generates pre-narrative, non-verbal video sequences resembling short sequences from the cinematic styles of silent film, compilation film, avant-garde cinema, documentary, music video, and home video.

Developing *Media Streams* required interdisciplinary research in artificial intelligence, film theory, and user interface design. The research in AI draws from work on dynamic memory, analogical understanding, and case-based reasoning (Schank, Lenat, Haase); the film analysis techniques borrow from formalist, structuralist, reader-response, and semiotic approaches (Metz, Eco, Bordwell, Iser), the work of Soviet silent film practitioners (Kuleshov, Eisenstein), and recent research on the aesthetics and practice of communities of television fans who appropriate and reuse found materials (Jenkins).

The thesis document is accompanied by a videotape with examples of video sequences retrieved/generated by the system.

Thesis Supervisor: Kenneth B. Haase

Title: Assistant Professor, Program in Media Arts and Sciences, MIT

Doctoral Committee



Thesis Advisor

Kenneth B. Haase
Assistant Professor
Program in Media Arts and Sciences, MIT

Thesis Reader

Seymour Papert
LEGO Professor of Learning Research
Program in Media Arts and Sciences, MIT

Thesis Reader

Henry Jenkins
Associate Professor
Literature and Film Studies Program, MIT

About the Author



Marc Davis has been a doctoral student in the Machine Understanding Group of the Learning and Common Sense Section at the Massachusetts Institute of Technology Media Laboratory since the fall of 1990. With a diverse background in literary theory, media technology, and artificial intelligence, he has researched and developed a prototype system called *Media Streams* for annotating, retrieving, and repurposing digital video. Marc Davis' work is about providing powerful tools for a whole generation of TV viewers (himself included) to make video a medium for global interaction and play. Marc Davis writes: "By putting the power of Hollywood and the networks in everyone's hands we will change the world."

Though working on research in artificial intelligence and multimedia, Marc Davis' education has been predominantly in the humanities. In 1984, he received a B.A. with high honors in College of Letters (an interdisciplinary program in history, literature, philosophy and language) from Wesleyan University in Connecticut. After completing a two-year research fellowship from the German Academic Exchange Service, he received an M.A. in literary theory and philosophy from the University of Konstanz in Germany. His areas of study included reader-response theory, deconstruction, postmodernism, fictionality, rhetoric, aesthetics, philosophy of language, phenomenology, and film theory. While at the University of Konstanz, he participated in, and computerized, the government-funded Intertextuality Research Project. Marc Davis also taught courses in improvisational dance, philosophy, and literary theory.

Combining his various backgrounds in humanities, arts, and media technology, Marc Davis co-founded (with Mike Travers) the Narrative Intelligence Reading Group at the MIT Media Laboratory. This weekly reading group explores issues at the intersection of literary theory, artificial intelligence, and media technology. Marc Davis has published papers and given talks on video representation, multimedia, virtual reality, interface agents, and user interface design. He has taught workshops and tutorials on museum exhibit design, constructivist learning, and interaction design. In 1992, he spent the summer interning at Mitsubishi Electric Research Labs in Cambridge, Massachusetts. Since the summer of 1993, Marc Davis has been completing his doctoral research while interning at Interval Research Corporation in Palo Alto, California. At AAAI-94 in Seattle, Washington, he co-chaired the Workshop on "Indexing and Reuse in Multimedia Systems" and at ACM Multimedia '94 in San Francisco, California, he chaired the panel on "No Multimedia Without Representation." Upon completing his doctorate Marc Davis will become a Member of the Research Staff at Interval Research Corporation and a Lecturer at Stanford University in Palo Alto, California.

Acknowledgments

Four and a half years ago a brand new Professor at the Media Laboratory, who had just recently completed his Ph.D. at the MIT AI Lab, took a chance on a crazed literary theorist who said he wanted to build computational media artifacts that could use analogy to make movies. That person was Prof. Ken Haase and without him, I would not have been a student at the Media Lab. I am profoundly grateful to him for his willingness to take a risk on me, support me, and to let me follow my research passions while sharing his own valuable insights and ideas.

The last four years I have had the incredible good fortune to have had the best undergraduate research assistants (UROPs) any graduate student could ever hope for. Golan Levin, Brian Williams, and I worked as a team sharing ideas, designs, insights, and many, many hours of inspiring work together. Golan and Brian define dedication, diligence, creativity, and a determination to bring forth the impossible at short notice. This thesis research would have been far far less than what it is without them. Thanks to Golan for his design genius and stamina in icon and interface design, for writing the Media Streams Manual, and for putting this document together. Thanks to Brian for his inspired and tireless brilliance in programming and documenting so much of the code of Media Streams, for his incisive problem-solving skills and technical wizardry, for editing and proofreading the thesis, and for assembling the supplemental videotape. Thanks to you both for all the work that went into the Media Streams User Study. After working together at the Media Lab for so many years, I am privileged and excited to continue working with you at Interval Research.

To my thesis readers—Seymour Papert who taught me to love the questions themselves and Henry Jenkins who continually amazes me with his erudition and dedication to the good fight—thank you for helping me along this journey by pointing the way with the example of your own lives.

My thanks also to the institutions and consortia that have supported this research (MIT Media Laboratory, BT, MoF, TVoT, NiF), to the two institutions that in addition to supporting my work through sponsorship of the Media Lab also offered me summer internships to pursue my research (Mitsubishi Electric Research Labs and Interval Research Corporation), and to the faculty who believed in my work and supported it with their counsel and their research funds (Ken Haase, Andy Lippman, Sandy Pentland, and Glorianna Davenport).

My thanks to the Narrative Intelligence Reading Group charter members who helped create such a vibrant locus of intellectual growth and comradeship during my years at the Media Lab: Mike Travers, Amy Bruckman, Jeremy Wertheimer, Carol Strohecker, Edith Ackermann, Henry Jenkins, Kathy Biddick, Margaret Minsky, Michael Johnson (a.k.a. “Wave”), Warren Sack, and Abbe Don.

A deep debt of gratitude goes to the colleagues at MIT who initiated me into the rites of hackerdom by teaching me how to program and think like a programmer: Mike Travers, Alan Ruttenberg, David Rosenthal, and Wave. Thanks to my serial officemates, Simson Garfinkel and Anil Chakravarthy, who shared with me the virtues of good programming and bad humor. Thanks to Barry Arons and Louis Weitzman for their friendship and help in preparing my thesis defense. Special thanks to Kevin McGee for his intellectual sparing and for sharing his thesis writing process with me before I had to go through mine. Thanks to all the other unmentioned fellow students at the Media Lab who helped make my experience there so rewarding. I'm looking forward to coming back for a demo.

I am very grateful to Interval Research Corporation for their support throughout this process. I want to especially thank the two mentors, now colleagues, who got me to come out to Interval: Brenda Laurel and Terry Winograd. Also thanks to other people at Interval who encouraged me to finish this dissertation by the promise of exciting collaboration ahead: David Liddle, Meg Withgott, Glenn Edens, Bonnie Johnson, Andrew Singer, Dick Shoup, Michael Naimark, and especially, David Levitt.

I would also like to thank the people at Interval who participated in the Media Streams User Study and the following people whose strategic, tactical, and logistical support made the User Study actually work: Golan Levin, Brian Williams, Bud Lassiter, Chris Seguire, Kathy Noesen, Shelly Wynecoop, Craig Hammond, Ellen Bewersdorff, and Diane Schiano.

I also want to express my gratitude to the teachers who taught me how to analyze and interpret texts: Wolfgang Iser, Renate Lachmann, Gayatri Chakravorty Spivak, and Judith Butler.

Special thanks to the artificial intelligence pioneers who taught me to appreciate the hopes, dreams, and difficulties of the discipline: Marvin Minsky, Seymour Papert, Roger Schank, Doug Lenat, R.V. Guha, Joseph Bates, and Terry Winograd.

Thanks to the two role models who taught me the meaning of wonder and integrity: Zippy the Pinhead and Jean-Luc Picard.

Deepest, heartfelt appreciation to my friends who saw me through this process with love, devotion, and favors far greater than I could have ever imagined: Jonathan Comras, Axil Comras, Jeff Goodfriend, Warren Sack and Jennifer Gonzalez, Abbe Don, Nancy Lipman, Mark Schneider, Eric Schneider, and Peggy and Grégoire Macy-Vion.

And finally to the person who, though in a medical residency program herself, found the time to love and support me through this process. All my love to you Wendy, always, and thanks, I'll make dinner now.



Table of Contents

0.

Title	1
Abstract.....	3
Doctoral Committee	5
About the Author.....	7
Acknowledgments.....	9
Table of Contents	11
Tables and Figures	17
Preface	21

1. Introduction: The Need for Video Representation **23**

1.1. The Problem	25
1.2. The Solutions	26
1.3. Scenarios of Representation and Retrieval	27
1.3.1. The Stock House	27
1.3.2. Garage Cinema	30
1.4. Technologies and Methodologies.....	31
1.4.1. Knowledge Representation	32
1.4.2. Film Theory and Analysis	34
1.4.3. Theory, History, and Practice of Visual Communications Systems	35
1.4.4. Techne-Centered Methodology	35
1.5. Significance of the Research	36
1.5.1. Knowledge Representation	37
1.5.2. Film Theory and Analysis	37
1.5.3. Theory, History, and Practice of Visual Communications Systems	38
1.6. Evaluation of the Research.....	38
1.7. Overview of the Thesis	40

2. Models of Video Practice **43**

2.1. Video Today.....	45
2.1.1. Video Annotation	46
2.1.1.1. Keywords	47
2.1.1.2. Why Keywords Are Not Enough	47
2.1.2. Video Archiving	48
2.1.2.1. Joe and Jane in the Basement	49

2.1.3. Video Editing	49
2.1.3.1. The Cutting Room Floor	49
2.1.4. Producers and Users	50
2.2. Video Tomorrow	55
2.2.1. Video Annotation	56
2.2.1.1. Video Archiving	56
2.2.2. Video Archiving	56
2.2.2.1. Joe and Jane and their Computers	57
2.2.2.2. Digital Storage	57
2.2.2.3. Emergent Archives	58
2.2.3. Video Editing	59
2.2.4. Producers and Users	59
2.3. Getting To Tomorrow	63

3. Media Streams Overview **65**

3.1. System Specification	67
3.2. Design History	68
3.3. Media Streams Functionality	68
3.3.1. Preprocessing	69
3.3.2. Annotation	69
3.3.3. Browsing	70
3.3.4. Retrieval	70
3.3.5. Repurposing	70
3.4. Media Streams Components	71
3.4.1. System Data Structures	71
3.4.2. System User Interface Components	72
3.5. Interaction Between Interface and Representation	76

4. Representing Video **77**

4.1. The Structures of Video	81
4.1.1. Basic Structures	81
4.1.1.1. Streams	82
4.1.1.2. Shot	82
4.1.1.3. Sequence	83
4.1.1.4. Scene	83
4.1.2. Representing Structures	84
4.1.2.1. Clips vs. Streams	84
4.2. The Functions of Video	87
4.2.1. Representational Status of the Video Image	88
4.2.1.1. Some Basic Concepts: Saussurean Linguistics	88

4.2.1.2. The Video Sign: Motivated or Arbitrary?	91
4.2.1.3. Cinematic Articulations: None or Three?	92
4.2.1.4. Video Image: Word or Sentence?	93
4.2.2. Video and Language	97
4.2.3. Video Syntax and Semantics	99
4.2.3.1. The Kuleshov Effect and Montage	99
4.3. Ontological Issues in Video Representation.....	102
4.3.1. Knowledge Representation for Video	103
4.3.2. Types of Video.....	103
4.3.2.1. Abstract Classification of Video Types	103
4.3.2.2. Historical Precedents for Non-Verbal, Action-Centric Video.....	105
4.3.3. Base Categories for Video Representation	106
4.3.3.1. Action	107
4.3.3.2. Character	111
4.3.3.3. Object	114
4.3.3.4. Relative Position	114
4.3.3.5. Screen Position.....	114
4.3.3.6. Mise-En-Scene: Space, Time, and Weather	115
4.3.3.7. Cinematography.....	117
4.3.3.8. Recording Medium	118
4.3.3.9. Transitions	118
4.3.4. Compositional Hierarchical Semantics	119
4.4. Automatic Representation	121
4.4.1. Video.....	121
4.4.1.1. Computable Low-Level Features.....	122
4.4.1.2. Computable Mid-Level Features	124
4.4.2. Audio	125
4.4.2.1. Pause Breaks	126
4.4.2.2. Specialized Audio Parsers	126
4.4.3. Why Automatic Representation is Not Enough	126
4.5. Representation for Retrieval and Repurposing.....	129

5. Retrieving and Repurposing Video **131**

5.1. Memory and Retrieval	134
5.1.1. Semantic and Episodic Memory	134
5.1.2. Media Streams Memory Structures	136
5.1.2.1. Framer	137
5.1.2.2. Example: Maya Lying on a Beach.....	141
5.2. Similarity and Retrieval	158
5.3. Repurposing and Retrieval	162
5.3.1. Media Streams Retrieval Mechanisms	163

5.3.1.1. Mnemosyne	163
5.3.1.2. Media Streams Retrieval Algorithm	164
5.3.1.3. Retrieval-By-Composition Example: How Did She Get There?	170
5.3.2. Repurposing and New Forms of Continuity	183
5.3.2.1. Retrieval-By-Composition Examples	185
5.3.3. Learning from Retrieval	193

6. Media Streams Interfaces 197

6.1. Visualizing Temporal Media	200
6.1.1. Spatializing Time	200
6.1.2. Media Streams Visualization Interfaces	202
6.1.2.1. Thumbnails	202
6.1.2.2. Videogram	203
6.1.2.3. Waveforms and Pause Bars	205
6.1.2.4. Media Time Line Icons	206
6.2. Annotation: Adding Structure	207
6.2.1. Precursor Notational Systems	207
6.2.1.1. Music Notation	207
6.2.1.2. Movement Notation	209
6.2.1.3. Storyboarding	210
6.2.1.4. Film Scoring	211
6.2.2. Annotation at Production Time	212
6.2.3. Making Descriptors in Media Streams	212
6.2.3.1. Icon Workshop	213
6.2.3.2. Extensibility of the Icon Language	223
6.2.4. Finding Descriptors in Media Streams	225
6.2.4.1. Icon Palettes	225
6.2.5. Making Descriptions in Media Streams	229
6.3. Browsing	235
6.4. Retrieval and Repurposing	238
6.4.1. Query By Description	239
6.4.2. Query By Example	239
6.4.3. Image Retrieval vs. Video Retrieval	239
6.4.4. Issues in Temporal Query	239
6.4.4.1. Temporal Duration	241
6.4.4.2. Temporal Scale	241
6.4.4.3. Temporal Relations	242
6.4.4.4. Continuity Relations	243
6.4.4.5. Relative Importance of Query Parts	244
6.4.4.6. Similarity Criteria	245
6.4.5. Media Streams Retrieval Interfaces	245
6.4.6. Query Refinement	245

6.4.7. Learning from Retrieval: The Analogy Editor	246
--	-----

7. Why Icons?	251
----------------------	------------

7.1. Iconic Visual Languages	253
7.1.1. Keywords vs. Iconic Visual Language	254
7.1.2. Natural Language vs. Iconic Visual Language	256
7.1.3. Text vs. Iconic Visual Language	257
7.1.4. Arguments Against Iconic Visual Languages	259
7.2. Media Streams' Iconic Visual Language	261
7.2.1. Icon Semantics and Syntax	262
7.2.2. Future Directions	263

8. Media Streams User Study	265
------------------------------------	------------

8.1. Motivation	267
8.2. User Study Questions	269
8.3. User Study Design	269
8.4. User Study Participants	276
8.5. User Study Results	278
8.5.1. Improving Annotation Rate	278
8.5.2. Re-use of Descriptive Effort.....	281
8.5.3. Convergence of Descriptions	284
8.6. Insights from the User Study	288
8.6.1. Interface Issues	288
8.6.2. Representation Issues.....	290
8.7. Learning to See in a New Way	291

9. Related Work	293
------------------------	------------

9.1. Knowledge Representation Approaches	298
9.1.1. Bloch: A First Attempt at Video Representation	298
9.1.2. Schank and His Students: From CD to CBR.....	298
9.1.2.1. Conceptual Dependency	299
9.1.2.2. Dynamic Memory	301
9.1.2.3. Case Based Reasoning.....	302
9.1.3. Lenat and Guha: Common Sense Knowledge Representation	303
9.1.4. Spatio-Temporal Logics	306
9.1.4.1. Spatio-Temporal Indexing	306
9.1.4.2. Settings	308

9.2. Integrative Approaches to Video Representation	310
9.2.1. Institute for System Sciences	310
9.2.2. MIT Media Laboratory	311
9.2.2.1. Cinematic Primitives for Multimedia and Stratagraph	312
9.2.2.2. BT Project	313
9.2.2.3. Domain-Specific Video Resequencing Systems	314
9.3. Comparison to Related Work	316
10. Conclusions and Future Directions	319
<hr/>	
10.1. Conclusions	321
10.2. Future Directions	322
10.2.2. Artificial Intelligence, Film Theory, and Media Technology	322
10.2.2. Towards Garage Cinema	323
References	325
<hr/>	
Appendices	345
<hr/>	
Appendix A: Media Streams User's Guide and Manual	345
A. One. Overview of the System Components	353
A. Two. Using the System	379
A. Three. Suggestions for Annotators	405
Appendix B: Code Listing for Media Streams Retrieval Algorithm.....	421
Appendix C: Media Streams User Study Games	439
Appendix D: Media Streams User Study Exit Questionnaire	447
Appendix E: Media Streams User Study Exit Questionnaire Results	453
Appendix F: Media Streams User Study Wrap-Up Discussion Transcript.....	467
Appendix G: Media Streams System Designers and Title Plate.....	483

Tables and Figures

Tables

Number	Title	Page
1	Current Production of Video in Five Market Sectors	51
2	Current Production of Video in Five Market Sectors	51
3	Future Production of Video in Three Market Sectors	60
4	Future Production of Video in Three Market Sectors	61
5	Matrix of The User Study Participants' Logging Rotation	273
6	The User Study Participants	276
7	Comparison of Perceived Difficulty of User Tasks	280
8	Comparison of Related Work	317

Figures.

Number	Title	Page
1	Current Production of Video in Five Market Sectors	52
2	Current Production of Video in Five Market Sectors	54
3	Future Production of Video in Three Market Sectors	61
4	Future Production of Video in Three Market Sectors	62
5	The Icon Space	73
6	The Media Time Line	73
7	Typical Work Flow in Media Streams Annotation	75
8	Horizontal and Vertical Segmentations of Video	81
9	Vertical Segmentation of Video	82
10	A Stream of 100 Frames of Video	85
11	Two "Clips" with Three Descriptors Each	85
12	A Stream of 100 Frames of Video with 6 Annotations	86
13	The Signifier and the Signified	89
14	Paradigmatic and Syntagmatic Structures	90
15	Constructed Continuity of Actor: A Kuleshov Effect	112
16	Continuity of Role in an Assembled Sequence	113
17	An Establishing Shot Sequence	116
18	A Two-Shot Elevator Sequence	118
19	FRAMER Structure for Fido the Wonder Dog's Legs	137
20	A Shot of Maya Deren Lying on a Beach	141
21	Constructing an Icon for Maya in the Icon Workshop	141
22	A Path Down the Character Action Hierarchy to <i>Lying</i>	144
23	A Path Down the Objects Hierarchy to <i>Beach</i>	145
24	A Path Down the Objects Hierarchy to <i>Wave</i>	145
25	A Path Down the Object Action Hierarchy to <i>Crash</i>	145
26	Paths Down the Spatial Location Hierarchy	146
27	A Media Time Line Describing the Shot from Figure 20	147
28	The 13 Temporal Relations	168
29	A Shot of Maya Deren Lying on a Beach	170

Number	Title	Page
30	A Media Time Line Describing Figures 29 and 20	170
31	A Cliff-Sea-Beach Diving Query	171
32	The Results of the Query in Figure 31	172
33	Stills Taken from the Best-Matching Result	172
34	The Score-Window for the Best-Matching Result	173
35	Stills from the Third Sequence Returned	173
36	The Score-Window for the Third Sequence Returned	174
37	Stills From the Fourth Sequence Returned	174
38	Stills From the Fifth Sequence Returned	175
39	The Score-Window for the Fourth Sequence Returned	175
40	The Score-Window for the Fifth Sequence Returned	176
41	Stills From the Tenth Sequence Returned	176
42	The Score-Window for the Tenth Sequence Returned	177
43	A Sea-Beach Diving Query	178
44	The Result Palette for the Query in Figure 43	178
45	Stills From the Best-Matching Sequence	178
46	The Score-Window for the Sequence in Figure 45	179
47A	A Ship-Diving Sequence Tied for Second-Best	179
47B	A Ship-Diving Sequence Tied for Second-Best	180
48A	The Score-Window for the Sequence in Figure 47A	180
48B	The Score-Window for the Sequence in Figure 47B	181
49	A Cliff-Beach Diving Query	181
50	The Result Palette for the Query in Figure 49	182
51	Stills From the Best-Matching Sequence	182
52	Duchamp's <i>Nude Descending A Staircase #2</i>	201
53	Delaunay's <i>The Eiffel Tower</i>	202
54	A Stream of Thumbnails of Video	203
55	Observing Shot Boundaries and Motion in a Videogram	204
56	A Videogram	204
57	Media Streams Represents Video at Multiple Timescales	205
58	The Audio Waveform and Pause Bars	205
59	Media Time Line Icons	206
60	Music Notation Before the Invention of the Staff	208
61	Modern Music Notation Makes Separate Parts Clear	208
62	Friedrich Albert Zorn's Dance Notation	209
63	A Storyboard With a Representation of Camera Motion	210
64	Storyboards Can Approach Comic Art in Design	211
65	Eisenstein's Film Score for <i>Alexander Nevsky</i>	211
66	The Icon Space	213
67	An Icon Path to <i>On Top of A Street in Texas</i>	215
68	The Subcategories of the Objects Hierarchy	218
69	The Two-Shot Elevator Sequence	222
70	The Icon Information Editor and the Icon Title Editor	223
71	The Animated Icon Editor	224
72	The Media Time Line	229
73	The Select Bar on the Media Time Line	230
74A	The Structure of the Video Annotation Streams	231
74B	The Structure of the Video Annotation Streams (Cont'd)	232
74C	The Structure of the Audio Annotation Streams	233

Number	Title	Page
75	The Eleven Movie Controls	236
76	The Minutes and Seconds Scrubbers	236
77	A Query of Three Annotations	240
78	Issue of Identity in Video Sequence Retrieval	244
79	The Search Control Palette	246
80	A Query for a Person Eating A Food-Object	247
81	The Results from the Query in Figure 80	247
82	The Analogy Editor	248
83	The Analogy Editor	248
84	The Results After Shifting the Prototype of <i>Shoe</i>	249
85	“Steve Biting a Dog” Situated in the Semantic Hierarchy	255
86	A Media Time Line Representing Jack’s Actions	256
87	A Stream-Based Natural Language Video Representation	257
88	Relevant Prior Experience of the User Study Participants	277
89	Annotation Performance of the User Study Participants	279
90	Perceived Difficulty of Media Streams Tasks	280
91	New Compounds Created Per Logger Per Session	282
92	Average New Compounds Created Per Session	282
93	Convergence Comparison of Participants’ Logs	284
94	Convergence Comparison of Experts’ Logs	285
95	Learning to See in a New Way	291
96	The Media Time Line	355
97	The Movie Controls	356
98	The Select Bar	356
99	A Non-Expandable Stream and an Expandable Stream	358
100	The Three Movie Streams	358
101	A Segment of a Seconds Thumbnails Stream	358
102	A Segment of a Videogram	359
103	The Minutes Scrubber and the Seconds Scrubber	359
104	The Cinematography Stream Controls	360
105	The Time-Index Displays	361
106	The Audio and Video Hidebars	361
107	The Elements of an Annotation	363
108	The Icon Space	365
109	Media Time Line Icons	370
110	The Icon Workshop	371
111	The Icon Information Editor and the Icon Title Editor	376
112	The Animated Icon Editor	377
113	An Icon Palette	378
114	Filter Units Can Return Media Time Line Icons	378
115	Typical Work Flow in Annotation	403
116	Functional Space Versus Scenery	410
117	Actual Versus Inferable Location	414
118	Different Objects with Identical Framings	420
119	The Media Streams System Designers	485

*Storm The Reality Studio.
And retake the universe.*
— William S. Burroughs, *Nova Express*

Preface

As a child I watched a lot of television. Hours upon hours of it everyday. And like millions of other American children I found that this activity, which took up so much of my time and interest, was not even discussed, let alone explored, analyzed, or taught in the other activity I engaged in for hours upon hours during the week: going to school. It was as if my entire culture was in a state of denial or suffered from widespread recurrent short-term amnesia. We learned how to write, how to read, how to speak, and manipulate numbers, but the skills that would have connected me to the affective center of my world—and to the engine of my culture's society and economy—were not taught. Today, children *and most adults* still do not have access to tools for creating, manipulating, analyzing, and playing with moving images and sound. My research is about radically changing that situation through the development of tools (both conceptual and computational) which will enable people to repurpose media (mass, popular, and personal).

Before beginning my doctoral work at the MIT Media Laboratory, I completed a Master's degree in literary theory and philosophy at the University of Konstanz in Germany. I learned how to analyze and interpret the structures and functions of literary and media artifacts, and the aesthetic responses of readers and viewers. In Konstanz, I also realized that if I had lived during the tremendous revolution in media technology brought about by the invention of the printing press, I would have left the academy to go work with Gutenberg. Today, one doesn't go to Mainz; one goes to Cambridge to work at the MIT Media Laboratory. At the close of the twentieth century, we are in the midst of an even greater transformation in media technology than of that from writing to print. We are transitioning to an era in which all information will be in digital form, and thus manipulable, transmissible, and sharable in ways that were never possible before in human history. In a few short years, we will live in a world in which large amounts of rich data (video, audio, text, numbers) will be able to be accessed, processed, and shared by people around the globe.

In order to make this vision a reality, the challenge which has to be addressed—and which many people both in industry and academia still ignore—is the representation of video content. For without a framework for representing the content of video information, without useful data about

video data, the dreams and hypes of the digital interactive media of the future will never come about. Researching and developing a framework for representing video content in order to support its retrieval and reuse has been the central theme of my graduate studies at the Media Laboratory. It has compelled me to bring together disciplines usually not on speaking terms: artificial intelligence, film theory, media studies, computer vision, semiotics, signal processing, reader-response theory, and user interface design.

The disciplinary divide between the humanities and the technical sciences is wide and deep. For example, when I first applied to the Media Laboratory, in an interview in the Visible Language Workshop, before I had received the statement of requirements for admission, I was asked, “What languages do you know?” I replied, “German, Latin, French...” After a stunned silence, my interviewers said, “No, what *programming* languages do you know?” The differences between humanists and technologists are more than a matter of differing vocabularies—they encompass fundamental concerns, methodologies, tools, standards of judgment, and social practices.

After teaching literary theory to first-year engineering students at Northeastern University and studying Lisp, C, and artificial intelligence, I began my doctoral career at the MIT Media Laboratory. There I became immersed in a culture of tool builders and hackers. While becoming one myself, I co-founded the weekly Narrative Intelligence Reading Group with Mike Travers. Its initial goal was to help Mike and me learn how to talk to each other by teaching each other our traditions through reading and interpreting their central texts and artifacts. It grew into a place in which researchers from inside and outside the Media Laboratory could articulate a common ground between literary theory, artificial intelligence, and media technology. After four years it is still meeting in the basement of the Media Laboratory.

This thesis has grown out of my childhood passions and a process of acculturation, transformation, and exchange through conversations, debates, tutelage, and long hours with colleagues at the Media Laboratory. It attempts to bridge the worlds of film theory, artificial intelligence, and media technology, and to create a set of ideas and the foundation for a technology which will put the power of a Hollywood studio, a network television station, and a vast film archive on everyone’s desk and in every kid’s garage.

It is my sincere hope that you will take the time to work and play with this thesis. Think of it as a power tool for the imagination.



Chapter One

The Need for Video Representation

Introduction: The Need for Video Representation



1.1. The Problem

Without content representation, the development of large-scale systems for manipulating video will not happen. Currently, content providers possess massive archives of film and video for which they lack sufficient tools for search and retrieval. For the types of applications that will be developed in the near future (interactive television, personalized news, video on demand, etc.) these archives will remain a largely untapped resource, unless we are able to access their contents. Without a way of accessing video information in terms of its content, a hundred hours of video is less useful than one. With one hour of video, its content can be stored in human memory, but as we move up in orders of magnitude, we need to find ways of creating machine-readable and human-usable representations of video content. It is not simply a matter of cataloging reels or tapes, but of representing the content of video so as to facilitate the retrieval and repurposing of video according to these representations.

Given the current state of the art in machine vision and signal processing, we cannot now (and probably will not be able to for a long time) have machines parse and understand the content of digital video archives for us. Unlike text, for which we have developed sophisticated parsing technologies, and which is accessible to processing in various structured forms (ASCII, RTF, PostScript, SGML, HTML) video is still largely opaque. Some headway has been made in this area. Algorithms for the automatic annotation of scene breaks are becoming more robust and enhanced to handle special cases such as fades (Otsuji and others 1991; Zhang and others 1993). Work on camera motion detection is close to enabling reliable automatic classification of pans and zooms (Teodosio 1992; Tonomura and others 1993; Ueda and others 1993). Problems which are still quite difficult but which are being actively worked on include: object recognition (Nagasaka and Tanaka 1992), object tracking (Ueda and others 1993), and motion segmentation (Otsuji and others 1991; Zabih and others 1993). Research is also being conducted in automatic segmentation and tagging of audio data by means of parsing the audio track for pauses and voice intensities (Arons 1993b), other audio cues including sounds made by the recording devices themselves (Pincever 1990), as well as specialized audio parsers for music, laughter, and other highly distinct acoustic phenomena (Hawley 1993). Advances in signal

separation and speech recognition will also go a long way to automating the parsing of the content of the audio track.

Yet this information alone does not enable the creation of a sufficient representation of video content to support content-based retrieval and manipulation. Signal-based parsing and segmentation technologies must be combined with representations of the higher level structure and function of video data in order to support annotation, browsing, retrieval, and resequencing of video according to its content.

1.2. The Solutions

The challenge is to develop usable technologies for the representation of video content that can leverage off of what machines can currently offer us and what humans can achieve with computational support. We are in need of technologies for representing video content which add structure to the signal such that video data becomes a structured data type which can more effectively support current functionality and uses, and more importantly, enable new uses and functionality.

This thesis is the description of such an effort to develop a representation language for video content.

We have invented a representation language for video content and a system that uses this language to enable humans to annotate, browse, retrieve, and resequence video data in ways which were not possible before.

We have developed retrieval algorithms for video content annotated in our representation language which make use of the structure and function of video as explicated by modern film theory and which utilize our current knowledge about the structures and functions of human and computational memory.

We have contributed to the repertoire of tools and techniques for visualizing and manipulating visual information by the integration and extension of several existing techniques into a coherent framework and by the invention of a new technique of representing and manipulating video content by means of iconic descriptors.

We have tested our working system with human users to verify the representational design. We have shown that the system is learnable, that users can make use of each other's descriptive effort, and that different users' descriptions of the same footage are semantically convergent.

We have built a system called *Media Streams* whose core functionality is to enable users to annotate and to *retrieve-by-composition* shots and sequences from an annotated archive of video streams based on user-formulated queries or annotated video segments. *Retrieve-by-composition* refers to the system's retrieval processes, which can either find an existing sequence in the annotated archive or compose a sequence from disparate segments in the archive in response to a user query.

The system was originally built for myself in order to solve a pragmatic problem of representing video content so that I could work on automatic movie generation. A necessary step in that research agenda has become this doctoral dissertation, yet this research and development effort promises to have more than a personal audience. If one were to ask who might use this annotation and retrieval technology, the answer would be that today it is for people who already spend time and money doing this, and that tomorrow, when digital video becomes a ubiquitous data type, it could be used by anyone interested in making movies.

1.3. Scenarios of Representation and Retrieval

Of the two communities that might use *Media Streams*, one currently exists and would be tremendously helped by the technology—archivists in stock footage houses. The other community is just beginning to exist and would grow and thrive by being able to use some future version of this technology as well as the footage annotated by the first user community—Garage Cinema makers.

Let us look more closely at scenarios of use within these two communities in order to understand the need for video representation and how the technologies developed in this thesis help answer that need, and enable new needs for video representation to be addressed.

1.3.1. The Stock House

Today's video archives exist in the vaults of news stations, television networks, video wire services, movie studios, and corporations. The footage they archive and retrieve is used for various video and film projects ranging from news and documentaries, to commercials, training films, and feature films. The function of the archive is to preserve footage and to retrieve it for use. At its core, a video archive stores footage that was created for one project or purpose and retrieves it for new projects and purposes. The whole idea behind a "stock footage house" is to have appropriate materials for making new films out of parts of old ones. In

other words, a stock footage house enables filmmakers to *repurpose* motion pictures, to use footage originally made for one purpose for new purposes other than those for which it was originally intended. A stock footage house is really a “repurposing house.”

Typically, stock footage archives operate using a hybrid of human and computational effort. A common retrieval process would involve a request from a human filmmaker for “a shot of x doing y in location z.” The archivist may then have access to a computerized archive of shots indexed by keywords. Even with such a computerized archive the process of retrieving the appropriate shots will largely rely on the memory of the human archivist. Keyword retrieval cannot retrieve shots based on *relationships* between the keywords, nor can it retrieve sequences of events or shots. Therefore, in order for the archivist to retrieve a specific shot or sequence, the relationships between the keyword terms which would describe the structured content of the video will either have to be remembered or found again by the archivist. Thus in stock footage houses today, even if the retrieval process begins with keyword based retrieval, it will end with the human improving both the *precision* and *recall* of the retrieval process.

The process of retrieving footage from the archive is *costly* and *time-consuming*. A typical retrieval request to a stock footage house will cost on the order of \$1500 per minute of footage purchased (plus library service charges of \$50 per hour of search and preparation time, duplication costs, and licensing fees) and a typical request will often take 48 hours turnaround time (CBS News Archives 1992). For the stock footage house, the labor and expense which go into meeting requests typically only result in a sale in one out of four or five requests:

To an unschooled eye, all fee schedules appear to be excessive. Yet few producers realize how much time stock footage libraries spend preparing footage for jobs that either dwindle in size or fail to materialize. In this author's experience, perhaps one out of every four or five research requests results in a sale. This explains why commercial libraries charge between \$25 and \$60 per hour for research time, though many research costs are not recovered and research fees are often waived in the case of bulk usage (Prelinger 1991).

The time, cost, and imprecision of current stock footage houses limit the use of the resources these archives contain for content producers and render them inaccessible to a possible community of desktop video makers, which in the years ahead will grow to include anyone using a computer for daily communication.

How would Media Streams change that process? Imagine a stock footage house that had annotated all of its footage with Media Streams. All of its content would be in digital form and indexed in such a way as to enable content-based retrieval over broadband networks. A human requesting footage could formulate a retrieval query either *by example* using similarly annotated footage they already have or *by describing* the desired footage with the annotation language of Media Streams.

The retrieval process would take seconds (as opposed to hours or days) to generate hits and through iterative query refinement the appropriate footage would be located. This process could be done by the human requesting the footage, by the human archivist, or by both humans working together.

Media Streams would increase the reusability of footage in the archive in several ways:

- by decreasing the time to find the footage
- by improving the precision and recall of the query
- by enabling the system to operate without the need of archivists to perform query and retrieval
- by increasing the usable content of the archive through retrieval-by-composition methods

By increasing the accuracy and speed of response, such a stock footage house could effectively outperform all of its competitors potentially resulting in a higher volume, lower cost, and broader market. Because of reduced cost and reduced search and retrieval time, the stock footage industry could broaden its market to include people who traditionally have not been able to regularly purchase stock footage. In addition to the obvious candidates for this broadened market (independent film and video producers, small TV stations, multimedia title makers, and interactive TV producers—who will need *lots* of program material), I am especially interested in a class of users/customers of stock footage that today is largely excluded from the production and distribution of motion pictures and sound: home video makers.

1.3.2. Garage Cinema

In the next five to ten years we will witness an explosion of access to and production of video by communities that could not earlier afford to produce video in their homes, schools, and offices. Just as desktop publishing gave consumers the power of the printing press on their desks (but it took the Net to make everyone a publisher since without it the distribution channel is lacking), and digital audio samplers gave birth to a whole new genre and population of music makers, desktop video technology will enable these and new communities to make video a part of their daily communication. In the spirit of garage bands, I think of this new population of motion picture producers as practitioners of “Garage Cinema.” It is what scratch, slash, rap, home video, and a TV, two VCRs, and a cable will become. These are the 15 year old kids who in 1998-2000 will be running a TV station/movie studio out of their garages.

In order for this to become a reality many technological, social, and legal changes have to occur, but the two major technological challenges that have to be met are the development of *tools for accessing content* and *tools for manipulating content*. The difference between a word processor and a Garage Cinema machine is that with language, if I want to tell you a story about a summer day in Paris in which a little dog stole my hat, I just did (or have begun to). With motion pictures I cannot simply speak or write images as I can sentences. In order to make my movie I have three options:

- I can take my production team to Paris or a studio and with several million dollars shoot and edit this story.
- I can wait 20-30 years for photorealistic computer graphics to become real-time and affordable.
- I can access stock footage of Paris, dogs, hats, even appropriate footage of actors, and piece together my movie.

For Garage Cinema makers the first challenge is getting access to appropriate footage in order to be able to tell a wide range of stories with footage that they cannot shoot or synthesize. That is the problem which Media Streams promises to solve by making stock footage available in a uniform and usable language designed for repurposing of footage and by making every videographer into a stock footage provider. The second problem of creating tools for manipulating content refers to the future challenge—partially addressed by Media Streams’ retrieval-by-composition

methods—of developing authoring tools which enable Garage Cinema makers to manipulate video according to its content.

There are of course already communities who engage in making Garage Cinema. With a TV, two VCRs, and a cable, fans of various television programs have for years been making their own movies out of found materials. This artistic and social practice has been studied by Henry Jenkins and reveals the ways in which fans refashion the audio-visual materials of popular culture into new motion picture artifacts which meet their own community's needs differently than the original material itself (Jenkins 1992).

Looking at fan video making practice and the use of samples in rap music one can piece together a vision of a Garage Cinema genre whose outlines we see today in the video making practice of teenagers who grew up with camcorders, VCRs, and computers. With tools for accessing content and tools for manipulating content, "home video making" may evolve into a genre in which home video adds to its expressive repertoire materials drawn from popular culture (movies, TV, news, cartoons, etc.). For example, imagine the type of video I could edit of my trip to the beaches, tropical jungles, and Mayan ruins of the Yucatan if, in addition to my own annotated footage, I had access to footage from documentaries about the Yucatan, the rain forest, and the Mayas, and from the television shows *Fantasy Island*, *Gilligan's Island*, the environmentally oriented cartoon show *Captain Planet*, and the "Masks" episode of *Star Trek: The Next Generation* (in which the Enterprise crew encounter an ancient ruin and culture similar to Pre-Columbian Central American civilizations).

The realization of technologies which promise to change and broaden the use of stock footage in the stock footage industry and in the communities of Garage Cinema makers is a task which requires methodologies of research and the integration and invention of technologies which do not fit solely within the boundaries of one discipline, but rather, which grow out of the boundary crossing and *bricolage* of several disciplines which historically have had little contact: knowledge representation, film theory and analysis, and the theory, history, and practice of visual communications systems.

1.4. Technologies and Methodologies

This thesis is about the representation, retrieval, and repurposing of video content. The research presented here combines methods and insights from three different fields: knowledge representation; film theory and history; and the theory, history, and practice of visual communications.

The problem of developing representation technologies for video content could not be solved within the intellectual domain and methodology of any one discipline. Creating a usable computational representation of video content requires that one combine technologies, ideas, and methods from several disciplines. The three main disciplines that come together in this research are knowledge representation, film theory and history, and the theory and history of visual communications systems. Each discipline contributes a series of *technologies* as well as unique *methodologies* of research and evaluation.

1.4.1. Knowledge Representation

Knowledge representation is one of the core subfields of artificial intelligence, which, in the years since its birth at the Dartmouth conference in 1951, has come to be situated within the disciplinary matrix of computer science departments. One of the key ideas which knowledge representation offers and which it inherits from programming language design is a set of criteria for constructing and evaluating computational languages:

A powerful programming language is more than just a means for instructing a computer to perform tasks. The language also serves as a framework within which we organize our ideas about processes. Thus, when we describe a language, we should pay particular attention to the means that the language provides for combining simple ideas to form more complex ideas. Every powerful language has three mechanisms for accomplishing this:

- primitive expressions, which represent the simplest entities with which the language is concerned,
- means of combination, by which compound expressions are built from simpler ones, and
- means of abstraction, by which compound objects can be named and manipulated as units. (Abelson and others 1985: 4)

The power that computational languages offer as opposed to natural languages is that they can be designed for particular types of expressivity, composability, and extensibility through abstraction. A language for representing, retrieving, and resequencing video can be judged according to the design of its primitive expressions, the means by which they can be

composed, and the means by which new expressions can be created through the naming of existing compound expressions. What computational languages provide is an extensible framework for describing and manipulating the structure and function of non-computational phenomena. It is precisely this power that current video systems lack and that future computational video systems must be based on.

Knowledge representation adds another key technological piece to the machinery of computational languages: *inference*. In addition to the ability to compose primitive expressions and make new expressions through naming of compound expressions, knowledge representation languages add the ability to construct *relationships* between expressions which support the automatic creation of new relationships between expressions through inference.

One of the most common forms of inference is known as *inheritance*. Inheritance is the relationship that enables properties of an expression to propagate to other expressions which *inherit* properties from it. This enables the language to minimize redundancy of expression because valid relationships between expressions can be implicitly as opposed to explicitly expressed. For example, rather than having to state that Fido the dog has four legs (and that for each dog I represent in the language that it has four legs), I can infer that Fido (and each future dog I will represent) has four legs, if it inherits the value of its number of legs either from the *prototypical* dog Spot to which Fido is *related*, or from the definition of the *class* of dogs of which Fido is an *instance*.

This Fido example points out two ways of organizing inheritance: a network in which new examples inherit features from prototypical examples; and a hierarchy of classes and subclasses in which new instances inherit their features from their class and superclasses. Either organizational strategy allows the creation of a semantic structure in which knowledge about the properties of things and their relations can be represented and inferred. Media Streams makes use of both structures of semantic representation: a hierarchy of classes and a network of prototypes.

Knowledge representation has also developed techniques for assessing the efficacy of representational systems. Some of the notions it contributes are ways of analyzing the *scalability* of a given representational system or method and attention to the inherent tradeoffs in representation, such as those between the precision and flexibility of a representation.

Finally, knowledge representation offers a unique paradigm for research into human mind, memory, and perception different from that used in psychology or philosophy. In attempting to understand how the human mind works, how our memories, perceptions, and feelings work, artificial intelligence researchers do not conduct experiments on humans or attempt to argue the validity of theories; instead, they construct and evaluate their ideas about structure and function by building artifacts which express their ideas. This methodology of constructing artifacts to explore and assess theories is the major methodological contribution of the discipline of knowledge representation to our research (I leave out here the strain of knowledge representation research that views formal theorem proving as an adequate research methodology). It has several important outcomes: the ability to play with and verify other's ideas through exploration of the artifacts they construct; the construction of artifacts which can be built on and improved on by others; and the creation of artifacts which may not only help elucidate, but may also augment and enhance human capabilities.

1.4.2. Film Theory and Analysis

Knowledge representation researchers have developed languages for describing physical and verbal systems. What they have not done is study the unique properties, structures, functions, effects, and history of cinema. That research program has been pursued by film theorists and analysts who have created a significant body of knowledge about cinematic artifacts.

Film theory and analysis take as their task the articulation of the structure and function of cinema. By viewing and writing about films, theorists try to explain how film, as a medium, works, and how given films "work." Film theorists have tried to understand the representational status and strategies of the cinematic medium by using various theoretical models which enable them to segment a film into its functional parts. Theorists invent ways to take films apart. The task of our research is to use this analytic knowledge in order to create artifacts which enable us to put films together. Film theory has built up an impressive array of theoretical constructs for their analysis of film ranging from definitions of the fundamental components of film shots and sequences, to models of cinematic time, action, narrative, narration, character, and point of view (to name a few). Unlike the constructivist methodology of knowledge representation, the predominant methodology of film theory and analysis is to apply (and sometimes create) theories by the analysis of existing artifacts.

1.4.3. Theory, History, and Practice of Visual Communications Systems

The third methodological and technological source for our research is a loosely affiliated grouping of several disciplines and practices which one could understand as “visual communications systems.” Under this rubric we bring together the theory, history, and practice of painting, graphic design, iconography, visual languages, and user interface design. The primary technological contributions to our research come from the semiotics and design of visual languages and the numerous techniques of representing time in space developed over the last 10,000 years though with special focus on the advances of the last century. The methodological stances of theorists and practitioners of visual communications systems run the gamut between the two methodologies described above, though our research is especially influenced by those technicians who build artifacts as a way of building theories.

1.4.4. *Techne*-Centered Methodology

In his excellent book on Sergei Eisenstein, David Bordwell describes what he considers to be the overriding aesthetic philosophy that guided the work of the early Soviet cinema pioneers as a “*techne*-centered philosophy.” Cutting across various avant-garde movements from Constructivism to Formalism was a common belief in the investigation of theoretical ideas through the construction of artifacts (practice) and the guidance of practical activity by theoretical concerns. Bordwell writes that “*Techne* is Aristotle’s term for the unity of theory and practice within skilled activity.” (Bordwell 1993: 35)

This methodological stance informed Kuleshov, Eisenstein, and a host of other early practitioner-theorists, i.e., technicians of cinema. The engineering methodology of knowledge representation research is much akin of the early Soviet avant-garde. In fact, many of the first filmmakers were trained as engineers and saw their task as a form of learning by making.

The work discussed in this thesis shares much methodological ground with the *techne*-centered poetics of the early Soviets. We argue for a *techne*-centered methodology for knowledge representation research on video and for film theoretic investigation of digital cinema. On the one hand, we stipulate that writing programs and building systems is an appropriate way

to create and test theories about cinematic structure and function. On the other hand, we argue that programs designed to analyze and assemble video which operate within the knowledge representation tradition need to be informed by the theoretical work of film theory and analysis. The work described in this thesis attempts to do both. Our techne-centered methodology takes up the project of the Soviet avant-garde within the technologies and methodologies of a computationally informed *constructionism*. This methodology was developed by Seymour Papert as a revisionist extension to Piaget's constructivist epistemology (Papert 1980). Aaron Falbel describes Papert's constructionism as follows:

Thus constructionism involves two types of construction: when children construct things out in the world, they simultaneously construct knowledge inside their heads. This new knowledge then enables them to build even more sophisticated things out in the world, which yields still more knowledge, and so on, in a self-reinforcing cycle. (Falbel 1993).

Our research seeks to construct artifacts through (de)constructing theories and to construct theories through (de)constructing artifacts.

1.5. Significance of the Research

The body of knowledge about how to represent video computationally and manipulate it at a level more significant than pixels or frames is at a very early, inchoate, and exploratory stage. This thesis will be one of the first comprehensive efforts in this research area and hopes to provide a basis for future work in the representation, retrieval, and repurposing of video content. On several counts this research signifies an original and significant contribution to knowledge:

- the creation of a memory-based representation utilizing both semantic and episodic memory structures for representing the sequence-independent and the sequence-dependent semantics of video content
- the development of representations and retrieval algorithms for video which take into consideration the semantics and syntax of the medium
- the development of a robust iconic visual language for the representation of video content

- the design of an annotation and retrieval system for video which is learnable, supports reusable description, and enables convergent descriptions of video content by various users

In addition to this written thesis, I have produced a working program, Media Streams, which enables users to annotate, browse, retrieve, and repurpose digital video.

This thesis is also an example of the type of interdisciplinary work required to solve the problems of a computational approach to video representation, retrieval, and repurposing that is informed by the insights of film theory. The hope is that this research will start a process of needed interchanges between film theory and knowledge representation that will support work in this new area.

The research presented in this thesis also has significance to the particular disciplines which inform its methodologies and technologies.

1.5.1. Knowledge Representation

This thesis is one of a few examples of the extension of the task of knowledge representation to the cinematic artifact. The pioneering work of Gilles Bloch (Bloch 1987) began this enterprise; our research represents a more comprehensive step toward the goal of representing video content, but much remains to be done in what one might think of as a new subfield of knowledge representation: knowledge representation for video (Davis 1994a). Within that subfield our research maps out the problems of representing video content, discusses ontological issues in video representation, critiques other efforts in this area, and proposes novel solutions.

1.5.2. Film Theory and Analysis

For film theory and analysis we extend the object and methods of the discipline by applying its techniques to a new problem: namely the computational analysis of video for the purposes of retrieval and repurposing. Furthermore, we propose a new methodology for the extension of film theory into the practice of building artifacts which analyze and compose digital video, namely the techne-centered approach discussed above which forms and validates theory through the construction of artifacts. We also hope to initiate a reevaluation and revitalization of an early period in film history (the Soviet Silent Era) so as to take up again the

project begun by Kuleshov and Eisenstein. Finally, we have created a testbed for exploring the efficacy of models of the structure and function of video sequences, especially those constructed from pre-existing parts.

1.5.3. Theory, History, and Practice of Visual Communications Systems

The significance of our research to visual communications systems is that we have constructed an artifact that enables us both to reevaluate the techniques of visual communications from the perspective of digital cinema, and to reevaluate multimedia interfaces from the perspective of the history of visual communications. One example here is our integration of existing video visualization techniques (thumbnails and videograms) into a framework that can be understood as extending the project of Cubism (Cooper 1970). Furthermore, we have created a large and robust iconic visual language for describing video that contributes new techniques of iconic combination, expression and search.

1.6. Evaluation of the Research

This thesis research combines methodologies and technologies from several disciplines. As interdisciplinary research, the question of how to evaluate the work brings the differences in approach between the disciplines into focus. One is confronted with several choices:

- evaluate the work within the context of only one of its disciplines and run the risk of making the results inaccessible to researchers in the other disciplines
- develop an entirely new methodology for evaluation of the work which has the merits of claiming new and well-suited intellectual ground, but runs the risk of being inaccessible to any researchers who may be interested in the work
- create a hybrid methodology for evaluation of the research that is at best soundly in the intersections between the various disciplines or at worst translatable from one methodological context to another.

In evaluating this thesis research I have chosen the third option.

The core of this research is within knowledge representation and is informed by work in film analysis and visual communications.

The methodology for evaluating a theory in knowledge representation, a subfield of artificial intelligence, is either to write a formal proof, which is only possible for knowledge representation schemes based on formal logic (which this work decidedly is not), or to build programs as a proof of concept of the theory, which has been the practice of evaluation since the inception of the discipline. The main evaluative question can be thought of as: does the program work?

In film analysis, practitioners test theories by analyzing the structure and function of particular cinematic artifacts according to the tenets of the theory and seek validation for the analysis within the community of film analysts and theorists. In many senses, the criteria for evaluation are rhetorical in film theory. The persuasiveness of an analysis or interpretation is the final arbiter of its validity within the interpretive community. The main evaluative question can be thought of as: does the analysis persuade?

In user interface design, the methodology of evaluation is far more heterogeneous than in either knowledge representation or film analysis. The entire question of evaluation is a research area within the discipline itself. The approaches cover a wide range: psychological experiments, ethnographic studies, aesthetic norms and judgments, audience research, reception aesthetics, theoretical correctness, participatory design with a community of users, iterative design, and building for one's own use.

The hybrid methodology I will use to evaluate the results of my research is as follows:

- the theory is instantiated by building a working program (knowledge representation)
- the program is an analysis of the structure and function of the artifacts it describes (film theory)
- the representational design of the theory/program is tested by having humans annotate video (user interface design)
- the retrieval mechanisms are evaluated on their aesthetic and cognitive coherence in retrieving and making new sequences (film theory and aesthetics)

Since the representations of video content in our system must be written and read by human beings in order to work, the methodologies of user studies needed to be applied in order to test the efficacy of the representational design. In order to determine whether the program “works” we conducted an extensive user study that set about answering the specific questions of the system’s learnability, the extent to which annotators will reuse each other’s descriptive effort, and the crucial question of whether different people’s descriptions of the same movies would be semantically convergent. The results of this study and the retrieval examples the system can generate should both serve to answer the question whether the theory/artifact we have constructed is a persuasive analysis of the structure and function of video that facilitates its retrieval and repurposing.

Finally, the ultimate evaluation of my thesis research will be within the community of researchers in the various disciplines it borrows from and operates within. If I succeed in influencing the theory and practice of some of these researchers, that will be the best evaluation of the thesis work.

1.7. Overview of the Thesis

Now that the basic premise of the research has been laid out, and its methodologies, claims, significance, and evaluation methods articulated, the remainder of the thesis will fill in and extend the scaffolding created in this introductory chapter.

In **Chapter 2**, we compare and contrast current video practice with models of future video practice that this research is based on.

In **Chapter 3**, we provide a brief introduction to and overview of the basic system components in Media Streams.

In **Chapter 4**, we delve into one of the core intellectual areas of the thesis research: the representation of video content.

In **Chapter 5**, we discuss the use of these representations in our retrieval and repurposing algorithms in detail.

In **Chapter 6**, we describe the main functionality, components, and contributions of the Media Streams user interface.

In **Chapter 7**, we argue for the use of our iconic visual language for describing video content.

In **Chapter 8**, we describe the design, execution, and results of our eight-person, two-and-a-half-day long Media Streams user study.

In **Chapter 9**, we outline and critique the related research in artificial intelligence and video representation that has contributed to the work discussed in this thesis.

In **Chapter 10**, we conclude with a discussion of the synergy of artificial intelligence research, media studies, and media technology that this thesis represents and point towards the eventual realization of the technological, social, and aesthetic changes this thesis hopes to initiate.



Chapter Two

Models of Video Practice

Models of Video Practice

2

2.1. Video Today

Current paradigms of video representation are drawn from practices which arose primarily out of “single-use” video applications. In single-use applications, footage is shot, annotated, and edited for a given movie, story, or film. Annotations are created for one given use of the video data. There do exist certain cases today, like network news archives, film archives, and stock footage houses, in which video is used multiple times, but the level of granularity, semantics, and non-uniformity to which these organizations annotate their archives limit the repurposability of this video content. The challenge is to create representations which support “multi-use” applications of video. These are applications in which video may be dynamically resegmented, retrieved, and resequenced on the fly by a wide range of users *other than those who originally created the data*.

Today, in organizations and companies around the world whose business it is to annotate, archive, and retrieve video information, by and large, the structure of the data is mostly represented in the memories of the human beings whose job it is to handle it. Even in situations in which keyword-based computer annotation systems are “used,” human short-term memory and long-term memory are the real repositories of information about the content of video data. “Joe and Jane in the basement” are the real indexing and retrieval mechanisms in almost all video archives. Human memory is very good at retrieving video due to its associative and analogical capabilities; it has memory structures which any computerized retrieval system would want to emulate. Nevertheless, there are significant problems in sharing the contents of one human memory with others and of transferring the contents of one human memory to another. There are also severe limitations in terms of storage capacity and speed for human memory that aren't acceptable if we are going to scale up to a global media archive in which video is accessed and manipulated by millions of people every day.

We need a language for representation of video content that enables us to combine automatic, semi-automatic, and human annotation so as to be able to make use of today's annotation effort long into the future.

2.1.1. Video Annotation

In developing a structured representation of video content for use in the annotation, retrieval, and repurposing of video from large archives, it is important to understand the current state of video annotation in order to create specifications for how future annotation systems should be able to perform. To begin with, we can posit a hierarchy of the efficacy of annotations:

At least, Pat should be able to use Pat's annotations.

Slightly better, Chris should be able to use Pat's annotations.

Even better, Chris's computer should be able to use Pat's annotations.

At best, Chris's computer and Chris should be able to use Pat's and Pat's computer's annotations.

Today, annotations used by video editors will typically only satisfy the first desideratum (Pat should be able to use Pat's annotations) and only for a limited length of time. Annotations used by video archivists aspire to meet the second desideratum (Chris should be able to use Pat's annotations), yet these annotations often fail to do so if the context of annotation is too distant (in either time or space) from the context of use. Current computer-supported video annotation and retrieval systems use keyword-based representations of video and ostensibly meet the third desideratum (Chris's computer should be able to use Pat's annotations), but practically do not because of the inability of keyword representations to maintain a consistent and scalable representation of the salient features of video content.

A video annotation language needs to create representations that are durable and sharable. The knowledge encoded in the annotation language needs to extend in time longer than one person's memory or even a collective memory, and needs to extend in space across continents and cultures. Today, and increasingly, content providers have global reach. German news teams may shoot footage in Brazil for South Korean television that is then accessed by American documentary filmmakers, perhaps ten years later. We need a global media archiving system that can be added to and accessed by people who do not share a common language, and the knowledge of whose contents is not only housed in the memories of a few people working in the basements of news reporting and film production facilities.

The visual language we have designed may provide an annotation language with which we can create a truly global media resource. Unlike other visual languages that are used internationally (e.g., for traffic signage, operating instructions on machines, etc. (Dreyfuss 1972) a visual language for video annotation can take advantage of the affordances of the computer medium. We have developed an iconic visual language for video annotation that is compositional, searchable, and extensible, and that uses color, shading, anti-aliasing, and animation in order to support the creation of durable and sharable representations of video content.

Rather than such a visual language, current video annotation systems typically use either free text or keywords. Given the current state of the art of natural language understanding, free text annotation is hopelessly unstructured for our purposes and even if we had perfect natural language systems, as we will discuss later, free text is unsuited to the task of describing video content. Keywords offer more structure and the promise of uniformity and computational tractability, but are also deficient for the task of creating sharable and durable representations of video content which support retrieval and repurposing.

2.1.1.1. Keywords

In the main, video has been archived and retrieved as if it were a non-temporal data type that could be adequately represented by "keywords." A good example of this approach can be seen in Apple Computer's *Visual Almanac* that describes and accesses the contents of its archive by use of "keywords" and "image keys" (Apple Multimedia Lab 1989).

2.1.1.2. Why Keywords Are Not Enough

This technique is successful in retrieving matches in a fairly underspecified search but lacks the level of granularity and descriptive richness necessary for computer-assisted and automatic video retrieval and repurposing. The keyword approach is inadequate for representing video content for the following reasons:

- Keywords do not describe the complex *temporal* structure of video and audio information.
- Keywords are not a *semantic* representation. They do not support inheritance, similarity, or inference between descriptors. Looking for shots of "dogs" will

not retrieve shots indexed as “German shepherds” and vice versa.

- Keywords do not describe *relations* between descriptions. A search using the keywords “man,” “dog,” and “bite” may retrieve “dog bites man” videos as well as “man bites dog” videos—the relations between the descriptions highly determine their salience and are not represented by keyword descriptions alone.
- Keywords do not *converge*. Since they are laden with linguistic associations and not a structured, designed language, keywords as a representation mechanism for video content suffer from the “vocabulary problem” (Furnas and others 1987). Different users use sufficiently different keywords to describe the same materials such that keyword annotation becomes idiosyncratic rather than consensual.
- Keywords do not *scale*. As the number of keywords grows, the possibility of matching a query to the annotation diminishes. As the size of the keyword vocabulary increases, the precision and recall of searches decrease.

Because of the deficiencies of keyword-based annotation and retrieval systems, video archives cannot rely on computers to overcome the inherent barriers to sharability and durability in human memory. In fact, even with today’s “computerized” systems video archives rely on human memory as the crucial repository of the knowledge not contained in computational representations.

2.1.2. Video Archiving

As described above, in current video archives it is human memory that contains the *temporal*, *semantic*, and *relational* information about video content. Because of this fact, video archives cannot achieve *convergence* or *scale up*. Current practice does not have the tools which would enable it to overcome these limitations.

2.1.2.1. Joe and Jane in the Basement

The real “intelligence” in current video archives are the men and women who work in them—Joe and Jane in the basement. It is in their memories and practice that the representations of video content are to be found. When a computer system is used to annotate footage it is not being used to create *representations of the content* of the video, but to create representations which serve as notes to jog the memories of the human archivists of that content or as pointers to footage in the archive which must be reviewed by a human in order to determine its content. That is what the “librarian hours” are for in a stock footage house’s bill: the time it takes an archivist to find the desired footage on the basis of the requester’s description, any reminders of or pointers to the contents of the archive, and the archivists’ own memories and work. Though this practice has worked for years it is widely perceived as too costly and inefficient. Without significant change, stock footage houses clearly cannot hope to meet the needs of supplying footage to everyone’s desk in the age of daily desktop video production and Garage Cinema.

2.1.3. Video Editing

Most current video editing practice and systems arise out of and exist within the context of *single-use* applications of video data. Footage is shot, logged, and edited for a single use, not for many possible reuses. Consequently, current editing systems do not build up or operate on representations of the content of the video data, but only on temporal markers of opaque data streams. Most editing systems represent video only by in and out points—both the representation of the video content and the knowledge of the constraints on the operations one could perform on this content are located in the mind of the editor. With such a cognitive load, video editing remains today a time-consuming and skilled craft.

2.1.3.1. The Cutting Room Floor

The traditional editing process is a matter of selecting and sequencing the appropriate shots for a given production (Rosenblum and Karen 1979). What does not get selected is usually discarded “on the cutting room floor.” Paradoxically, what gets left on the cutting room floor may be the raw material that would give someone else the leverage they would need to repurpose the content. The out-takes from a production may, for this reason, be valuable for they often articulate or at least indicate roads not taken. We see evidence of this today in the interest of interactive multimedia title producers who seek to obtain rights to the out-takes of Hollywood movies.

2.1.4. Producers and Users

In order to understand the current technologies for video representation, retrieval, and repurposing it is useful to consider the relationship between the contexts in which people produce and use video content. Understanding how the relationships between the contexts of production and use inform the design of current technologies (and vice-versa) may enable us to speculate on how the technology we have developed may inform and be informed by its possible contexts of production and use.

In considering relations of production and use, one issue stands out most markedly: the pervasive absence of editing and distribution technologies outside the hands of a few highly trained professionals and the corporations and institutions which fund them. This concentration of technologies of authorship and publication that exists for video today is analogous to the centralized control of printing and publication technologies in the early phases of their development. Our hope is that as word processing and the Net have made widespread decentralized authorship and publication of text possible, the technologies we have developed will contribute to a similar change in the relations of production and use for video.

Let us consider the differences in production and use of video in five market sectors:

- Hollywood
- Television Networks
- Corporate Audio-Visual Communications
- Independent Film and Video Producers
- Home Video Makers

In the table below we compare for each of these market sectors the following features:

- the total number of people in the sector involved in production
- the number of people who typically use an *individual* production
- the relative training (professionalism) of people in the sector involved in production
- the total cost in the sector of individual productions
- the periodicity of typical use in the sector of individual productions

The figures in the table are rough approximations and can be thought of as accurate to plus or minus one order of magnitude. Hence the important differences are those which separate features by two orders of magnitude or more.

Table 1.

Market	Producers	Users	Training	Cost/Production (\$)	Time
Hollywood	10,000	10,000,000	High	10,000,000	periodic
Networks	10,000	100,000,000	High	1,000,000	daily
Corporate AV	1,000	10,000	Medium	10,000	periodic
Independents	1,000	10,000	Medium	100,000	periodic
Home Video	1,000,000	10	Low	100	periodic

In order to help us visualize the relationships between the features of the five market sectors we can reduce the five features to four by collapsing the features of training and the cost per production into one feature since they are closely correlated. Our data would then look as follows:

Table 2.

Market	Producers	Users	Training & Cost	Time
Hollywood	10,000	10,000,000	10,000,000	periodic
Networks	10,000	100,000,000	1,000,000	daily
Corporate AV	1,000	10,000	10,000	periodic
Independents	1,000	10,000	100,000	periodic
Home Video	1,000,000	10	100	periodic

The visualization of this data below plots each market sector by a circle. All axes are scaled logarithmically. The X axis is the total number of people in the sector involved in production. The Y axis is the number of people who typically use individual productions. The collapsed and correlated features of the relative training (professionalism) of people in the sector involved in production and the total cost in the sector of individual productions are represented by the value in the sector's circle. The smaller this value, the more amateur the producer and the less expensive the cost of production in the market sector.

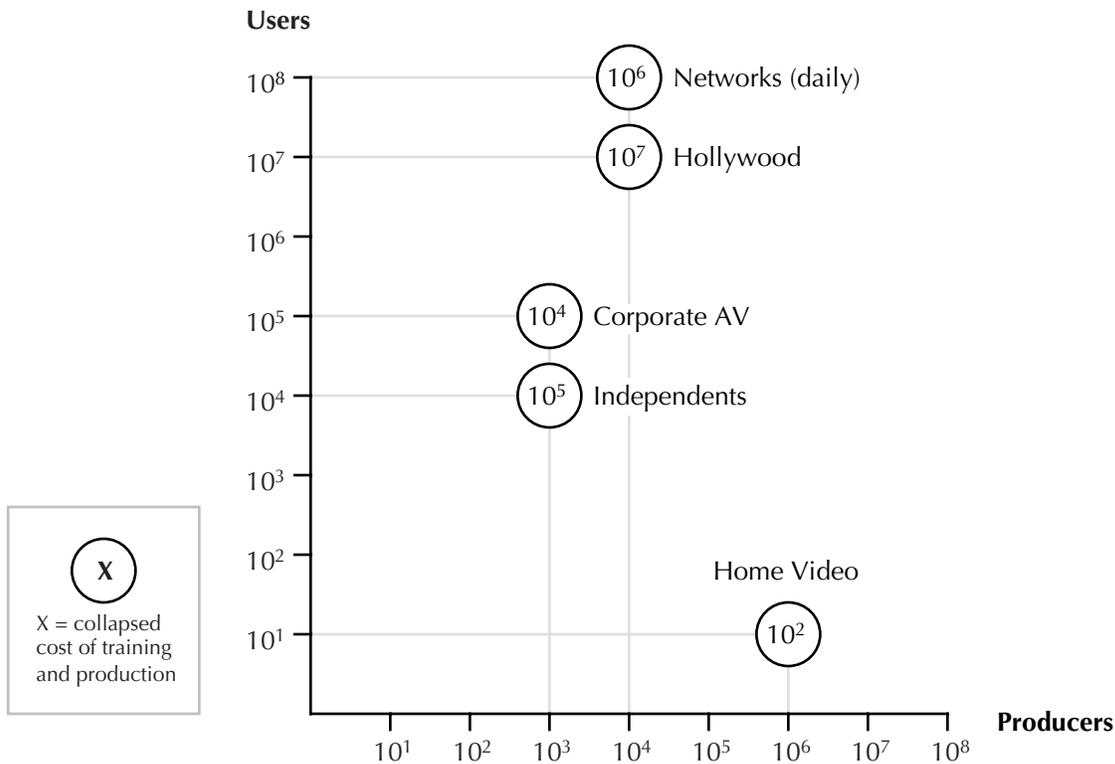


Figure 1. Video Production in Five Market Sectors

The above visualization reveals interesting correlations in the data. The first thing that stands out is the clustering of various market sectors: Hollywood and the networks; corporate AV and independents; and home video off by itself. The distribution of these clusters tells us something as well. The lower left and upper right quadrants of the plane formed by the axis of users and the axis of producers are unpopulated. The reasons for this are intriguing. A market sector that inhabited the lower left quadrant or the upper right quadrant would have a close parity between the number of users and producers. In the lower left quadrant each population would be small; in the upper right quadrant each population would be quite large. The absence of market sectors inhabiting these quadrants reveals to us the past and the future of the relationships between users and producers: an 18th century town meeting in a town hall would map to the lower left quadrant; a 21st century electronic town meeting on a multicast network would map to the upper right quadrant.

There is also a correlation in the respective clusters between the position of market sectors in the plane formed by the axis of users and the axis of producers and their position on the axis of the correlated training of producers and cost of production. In order to conceptualize this

correlation, let us first create a *ratio* between users (the number of people who typically use individual productions) and producers (the total number of people in the sector involved in production). It is important to note that the size of the typical population of users for any given production is not the same as the size of the total population of users (except in the case of a total broadcast situation like an All Points Bulletin on police radio in which these populations would be the same). The size of the total population of producers is not the same as the size of the population of producers involved in any given production (a case in which this would be true would be the biggest rock concert imaginable in which all musicians in the world would take part—coming to InterNet Radio any day now).

The users/producers ratio represents the symmetry or asymmetry in the relationship between users and producers. The value of this ratio can help us distinguish different models of the users/producers relationship.

A users/producers ratio equal to 1 corresponds to:

an **intercom** model (all-to-all—the typical number of users of individual productions is *equal to* the total number of producers).

Higher values correspond to:

a **broadcast** model (few-to-many—the typical number of users of individual productions is *greater than* the total number of producers).

Lower values correspond to:

a **newsgroup** model (many-to-few—the typical number of users of individual productions is *less than* the total number of producers).

Let us visualize the correlation between the users/producers ratio and the correlated training of producers and cost of production by plotting the various market sectors using this users/producers ratio as the X axis and the correlated training of producers and cost of production as the Y axis (both axes are logarithmic).

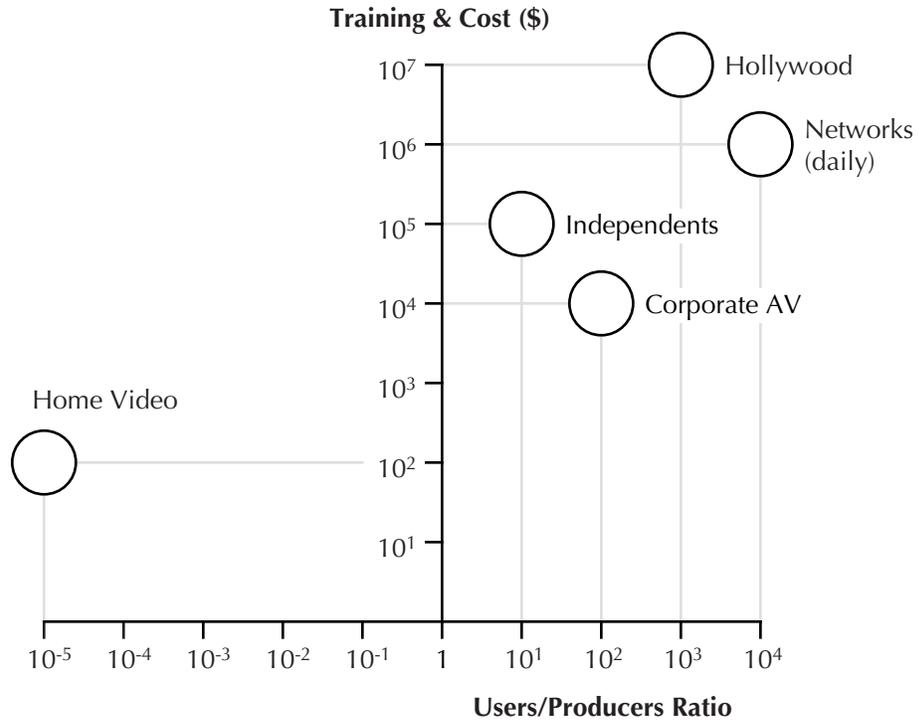


Figure 2. Video Production in Five Market Sectors

As one might expect the visualization shows that as market sectors require more training of producers and the costs of production increase, the relationship between users and producers changes from a low users/producers ratio (newsgroup model) to a high users/producers ratio (broadcast model).

The second visualization shows the tremendous distance between high end production and home users (both in the users/producers ratio and in the correlated training of producers and cost of production). The first visualization reveals the intriguing absence of market sectors in the upper right quadrant of the plane of users/producers.

The opportunities for future technologies of production and use (specifically, for the technologies we have developed in our research and the technologies which may be developed with them) are twofold: to close the gap between Hollywood and the home; and to move technologies, users, producers, and markets to the upper right quadrant of Figure 1.

2.2. Video Tomorrow

In the near future, we can imagine a world in which video annotation, search, and retrieval are conducted not just by professionals for professionals, but by anyone interested in repurposing footage. In a world where digital media are produced anywhere by anyone and are accessible to anyone anywhere, video will need to accrete layers of content annotations as it moves around the globe throughout its life cycle of use and reuse. In the future, annotation, both automatic and semi-automatic, will need to be fully integrated into the production, archiving, retrieval, and reuse of video and audio data. In production, *data-cameras* will encode and interpret detailed information about where, when, and how they are recording and attach that information to the digital data stream. Global satellite locators will indicate altitude, longitude, and latitude, while time will be stamped into the bit stream. Other types of sensing data (temperature, humidity, wind, etc.), as well as data about how the camera moves (pans, zooms, etc.) and how far away the camera is from its subjects (e.g., range data) will provide useful layers of annotation of the streams of video and audio data which the camera produces (Davenport and others 1991). In situations other than field recording (e.g., motion picture production), data from storyboarding, previsualization, production, and editing will be integrated into an environment in which all that is known about the footage being produced forms a set of coherent annotations of the video stream (Lasky 1990).

In indexing, humans will be able to annotate and index incoming footage using a robust and sharable annotation language that helps them annotate new footage by analogy to already annotated and indexed footage. In retrieval, analogically indexed annotations will enable people to manipulate and retrieve footage by similarity to existing footage (think of it as the implementation of a button with which the user can say "get me something *like that*"). Either by analogy to a given video sequence or by the specification of an abstract query, the user will be able to retrieve video sequences that can be assembled on the fly from a large archive of annotated footage.

By having a structured representation of video content—meaningful bits about the bits—future annotation and retrieval technology will enable users to mix video streams according to their contents and to manipulate video at various levels of granularity. With this kind of representation,

annotation, and retrieval, we will have tools which enable users to operate on higher-level content structures as opposed to being stuck with just bits, pixels, or even frames or clips.*

2.2.1. Video Annotation

There will remain many other annotations of a more semantic nature which future data-cameras won't be able to automatically encode, and for which we will want to have formats such that humans working with machines will be able to easily write them. In a sense, the challenge is to develop a language of description which both humans and computers can read and write and which will enable the integrated description and creation of video data. Such a language would satisfy the fourth desideratum of video annotation (Chris' computer and Chris should be able to use Pat's and Pat's computer's annotations).

2.2.1.1. Video Archiving

In order to overcome the inherent limitations of current keyword-based systems, we need to develop representations which capture the *temporal*, *semantic*, and *relational* content of video. These representations also need to be *convergent* and *scalable*. We have developed a language for the description of video content which addresses these issues. Chapters 4 and 5 detail the design of our solutions.

2.2.2. Video Archiving

The most significant change in video archives in the future will be the extent to which humans and computers will be able to work together in order to overcome the limitations of human-only and computer-only systems. Future video archives will contain representations which are durable and sharable, which converge and scale, and which make possible the creation, maintenance, and growth of a global media archive.

* The recent spate of Miller beer commercials provide an interesting vision of this form of future control over video at the level of its content. In an editing style one might call "conceptual morphing" two separate TV channels are blended into one. A channel showing a luge competition and one showing a bowling competition become one showing "luge bowling." Another example conceptually morphs Sumo wrestling and high diving, yet another combines drag racing and a dog show.

2.2.2.1. Joe and Jane and their Computers

Joe and Jane are no longer trudging around the basement. They, and millions of other people, are working in a way that combines human and computational memory, that leverages what humans are good at and what computers are good at, and that uses a common annotation language legible and writable by humans and computers.

Joe and Jane's computers are also different. They do not attempt to solve all problems on their own. Designers of signal-processing algorithms and designers of human-computer interfaces have finally found ways to work together such that a whole new class of "human in the loop" algorithms and systems have been developed that enable humans and computers to use each other's strengths in order to solve problems neither of them could on their own.

2.2.2.2. Digital Storage

The future will be digital. The core technology that will enable the rapid analysis, annotation, browsing, retrieval, and repurposing of video is digitization. Future video archives will have all of their footage in some digital form. It will either be stored in full resolution digital format or in a proxy or surrogate low resolution digital format useful only in an off-line mode. The distinction between off-line and on-line modes comes from video editing (Rubin 1991). On-line refers to when the actual materials to be edited are being manipulated. Off-line refers to when surrogate or proxy versions (often lower resolution copies) of the materials are being manipulated. In an off-line mode the manipulations on the surrogate materials result in the specification of a list of operations to be performed on the primary materials (often called an *edit decision list* or EDL). Because of the expense of full resolution digital storage, most stock footage houses can only justify the storage of low resolution digital proxies of full resolution analog footage (Kornel 1992). When the economics of storage, retrieval, and distribution change, this situation will also change. Currently, the dollar to gigabyte ratio stands at approximately \$400/gigabyte (sharply down from last year's \$900/gigabyte). A single frame of uncompressed NTSC quality video is approximately 1 megabyte. Assuming 30 frames per second and 30:1 compression, a single hour of compressed NTSC video will use up 3.6 gigabytes of storage (1 megabyte per second x 60 seconds per minute x 60 minutes per hour). MPEG compression reduces this figure to a half or a third of the amount, but with a sometimes noticeable loss in video quality. If we assume existing compression and cost ratios, a 1000 hour video archive would require 3.6 terabytes of storage and cost \$1,620,000. If we assume that in four years

disk prices will have roughly quartered (\$100/gigabyte) and that high quality MPEG compression will be standard (3 megabits/second), then that same 1000 hour video archive will require only 1.35 terabytes of storage and cost a mere \$135,000. If archival quality videotape costs roughly \$13.50 per tape today, then in four years digital storage will be only one order of magnitude more expensive than today's conventional storage media.

Video archives of the future will not only be digital, but *networked*. This network of digital video archives will form a global media resource containing footage from Hollywood movie vaults to home videos stored in basement servers around the world.

2.2.2.3. Emergent Archives

An archive is traditionally conceived of as a central repository of documents or objects administered and indexed by highly trained professionals working at one central site. Thus the materials in the archive can be preserved, guarded, catalogued, and accessed by experts. The Net is currently bringing about a very different kind of archive, what Kathy Biddick refers to as an "emergent archive" (Biddick 1994).

Biddick's emergent archives have no central locus of control or repository of objects. Anyone who has traversed the World Wide Web has experienced such an archive first hand. Video archives of the future exhibit many of the characteristics of emergent archives as described by Biddick. What distinguishes video archives from text archives is the need for annotation of their contents in order that they can be archived at all. Video and audio data require additional descriptors so that they can be made accessible to the type of retrieval that can make use of their semantic content and structure. The hypothesis in Media Streams is that adopting a uniform scheme of annotation for representing those aspects of video content which support its repurposing will be of benefit across contexts and users. Therefore the emergent video archives of the future may be emergent in their physical structure and topology, but in order to have semantic and granular access to their contents, emergent archivists would need to adopt a common markup scheme. In a sense, HTML itself is a good example of such a necessary, ubiquitous, and uniform markup language whose functionality makes emergent archives of text possible. Similarly, Media Streams may be the kind of linchpin technology that enables distributed video archives to emerge which are created and administered around the world in corporations, universities, basements, and garages.

2.2.3. Video Editing

With large repositories of annotated footage we will be able to develop applications that make use of their representations. Editing will become more like storyboarding in that a description of a movie will then generate a movie. Media Streams' retrieval-by-composition methods are a step in this direction.

These new editing tools which make use of content representations will complete the answer to the two major needs of Garage Cinema makers: tools for accessing content and tools for manipulating content.

2.2.4. Producers and Users

In the next decade, the emergence of a global media archive and tools which can access and manipulate this archive according to its contents will enable fundamental changes in the relationships between producers and users of digital media.

The existing five market sectors we examined before—Hollywood, Television Networks, Corporate Audio-Visual Communications, Independent Film and Video Producers, and Home Video Makers—will have changed.

On the high end of video production, Hollywood and the Networks have merged into a series of entertainment conglomerates producing content for various digital venues. Let us refer to these future high-end content producers as *Channel Hollywood*. Though audiences have fragmented due to the hundreds of specialty broadcast channels and the explosion of the Net, with its millions of video servers containing user-produced video content from the home and the workplace, the ability to have a common viewing experience, threatened by these events and the waning of movie theaters, is still desirable. The broadcast modes of Channel Hollywood are still popular due to their socio-cultural function in bringing people together to watch the same things at the same time as movie houses and traditional television networks once did.

Corporate AV is an even more active market sector than before. Video has become a ubiquitous data type used by many people in daily communications, reports, design, memos, etc. Just as word processing software and spreadsheets brought about the decentralization of accounting and document preparation functions away from a core few specialists to most knowledge workers in a corporation, video annotation, retrieval, and repurposing tools have resulted in the relocation of many of the functions of corporate AV departments to people's desktops.

The most profound changes have occurred at the lower end of video production. Changes in technology have brought about a merging of independent video producers and home video makers into a broad and active market sector we call *Garage Cinema*. Today people speak of the “New Hollywood” and refer to the merger of Hollywood and Silicon Valley. When the tools and infrastructure are in place to enable cheap and effective home use of video annotation, retrieval, and repurposing tools, the garages of the world will be the sites of the “New New Hollywood.” The conditions of production and use will have changed such that a large group of amateurs will be regularly making video with very low production budgets that can compete in the information marketplace of the Net.

The television networks have been supplanted by a situation in which the Net works. The new technologies of production and distribution will enable the commercialization of amateur and home work resulting in an explosion of cottage industries of video annotation and production. The beginnings of on-line distribution of video footage are already visible today in such recent ventures on the World Wide Web as FOOTAGE.Net (<http://www.footage.net:2900/>). As the PC revolution of the 1980’s brought the text and numerical processing power once held by corporations to people’s desktops, in the next decade the production and distribution power of Hollywood studios, television networks, and stock footage houses will reside on people’s desks and in their garages.

The challenges facing this future are not so much technological as legal. In the context of non-commercial use of stock footage the fair use doctrine needs to be clarified for video (Siporin 1990) and in order for for-profit Garage Cinema to “work” we need an economic and legal system that streamlines the sale of stock footage to low-cost producers (Thomson 1988). Technological, legal, and economic structures such as digital access, digital copyright, and digital billing can change the situation for stock footage producers and users so that video content can be widely sold and resold for reuse.

We can perform the same analysis of these three new market sectors as we did on the five current ones before:

Table 3.

Market	Producers	Users	Training	Cost/Production(\$)	Time
Channel Hollywood	10,000	100,000,000	High	1,000,000	daily
Corporate AV	10,000,000	10,000	Low	100	daily
Garage Cinema	10,000,000	100,000	Low	100	daily

We can perform the same feature reduction as before, by collapsing within sectors the cost of professional training and the cost of individual productions:

Table 4.

Market	Producers	Users	Training & Cost	Time
Channel Hollywood	10,000	100,000,000	1,000,000	daily
Corporate AV	10,000,000	10,000	100	daily
Garage Cinema	10,000,000	100,000	100	daily

Visualizing these features and market sectors in the same way as before we obtain the following graph:

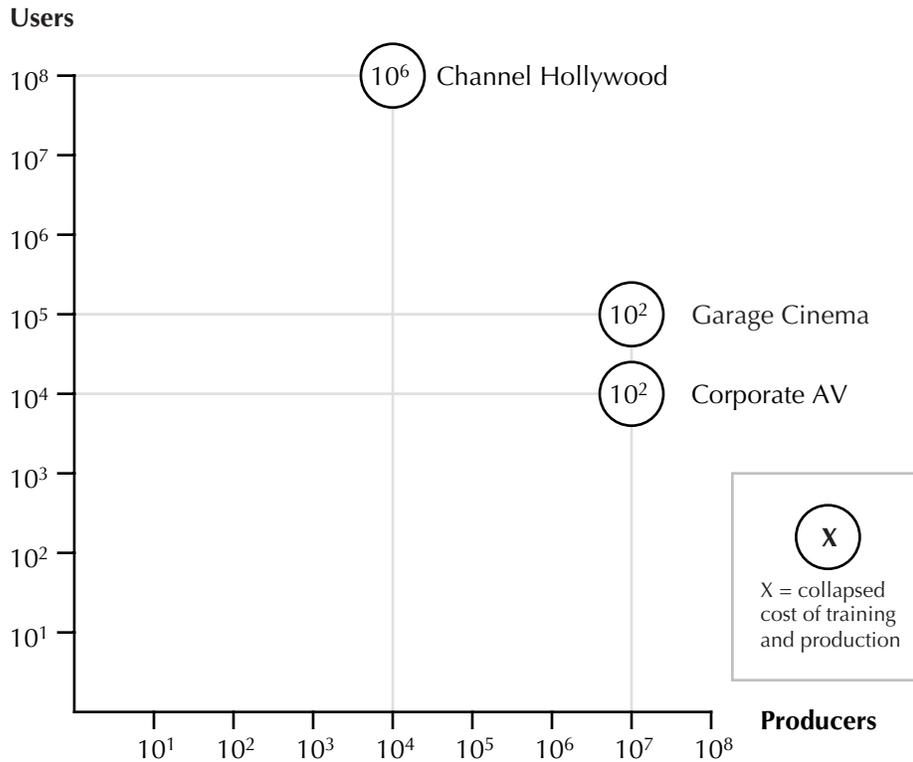


Figure 3. Video Production in Future Market Sectors

The above visualization again reveals interesting correlations in the data. Now there are new clusterings of market sectors: Corporate AV and Garage Cinema; and Channel Hollywood off by itself. The new distribution of

these clusters tells us something as well. Corporate AV and what were independent film and video producers and home video (now Garage Cinema) have migrated towards the upper right quadrant, towards a greater symmetry of a greater number of people in the production and use of video content.

The visualization below of the relationship between the users/producers ratio and the correlated training of producers and the cost of production corroborates and refines this result.

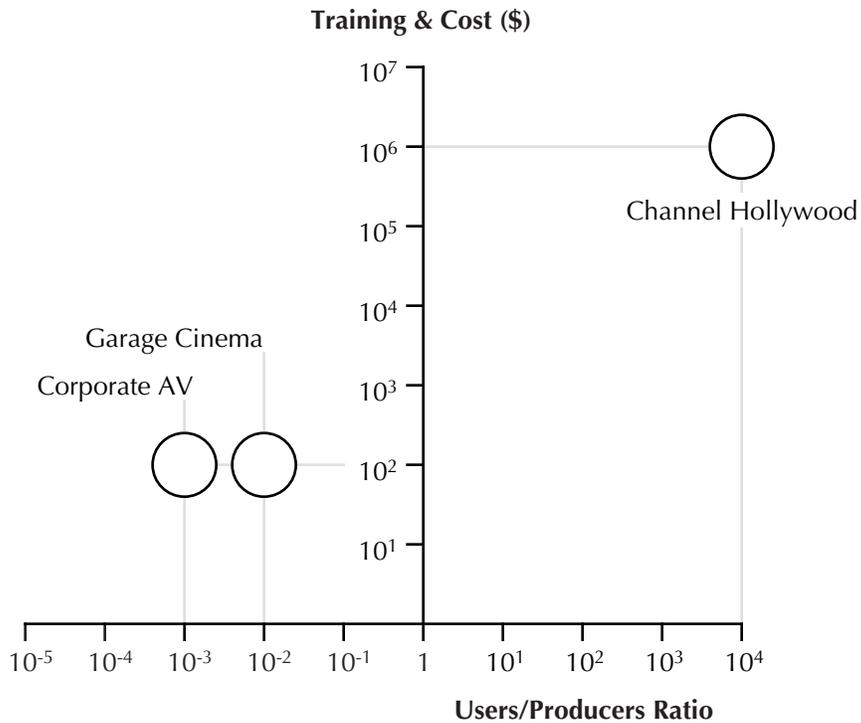


Figure 4. Video Production in Future Market Sectors

Corporate AV has flipped sides on the graph because of the ubiquity of video authoring tools and its ascendancy as a common data type for daily synchronous and asynchronous communication. It has also dropped lower on the training and cost of production axis since video production is now a low cost, non-professional, daily activity. The users/producers ratio for both Corporate AV and Garage Cinema have moved closer both to each other and to unity, closer to the symmetry of producers and users of an intercom model, but still retain the beneficial properties of the newsgroup model. The asymmetry of the newsgroup model allows for communities of choice and affiliation to form which would not be possible in the technologically re-enforced single community of the intercom model. The

perfect symmetry of the intercom model does not scale to large, diverse communities because of the lack of boundaries. Living in a perpetual town meeting would mean the death of privacy.

Hollywood is now farther from Corporate AV and still quite distant from the home (Garage Cinema). Channel Hollywood now faces the challenge of millions of Garage Cinema producers and users who operate in an economy similar to the newsgroups of today in that they appeal to and form highly specialized interests and communities. The interplay between high end broadcast cultural production and convivial newsgroup style cultural production is more active than ever as users have become producers in a global media funhouse which people used to call the information superhighway.

2.3. Getting To Tomorrow

How do we get from video today to video tomorrow?

There are many challenges which need to be addressed. The necessary but not sufficient condition we have tried to fulfill is to develop a language for the representation of video content that enables users *to annotate, retrieve, and repurpose video sequences*.

The description of the particular technological challenges and solutions we have developed occupies the next five chapters.



Chapter Three

Media Streams Overview

3. Media Streams Overview

3

3.1. System Specification

Over the past four years, a small group of us in the MIT Media Laboratory's Machine Understanding Group (myself with the assistance of Brian Williams and Golan Levin under the direction of Prof. Kenneth Haase) has built *Media Streams*, a prototype for the representation, retrieval, and repurposing of video and audio data (Davis 1991; Davis 1993a; Davis 1993b; Davis 1994a; Davis 1994b).

Media Streams is written in two languages: the outstanding rapid prototyping environment of Macintosh Common Lisp (Apple Computer 1993a) with its CLOS (Common Lisp Object System) interface to the Macintosh ToolBox, and FRAMER (Haase 1994), a persistent framework for media annotation and description that supports cross-platform knowledge representation and database functionality. FRAMER was conceived of and developed by Prof. Kenneth Haase and offers the type of flexible and extensible multimedia database and knowledge representation language we wish commercial vendors could provide us with.

In addition to using the many extensions to the Macintosh Common Lisp base system contributed by hackers at the Media Lab (Alan Ruttenberg being the major contributor among us) and the incredible community of MCL hackers around the world, we also invented many new interface widgets ourselves, and, with the help of Mike Travers (MIT Media Lab Machine Understanding Group) and colleagues at the Institute for the Learning Sciences at Northwestern University, extended Macintosh Common Lisp's interface to the Macintosh ToolBox by writing a CLOS interface to Apple's QuickTime (Apple Computer 1993b), a digital video format that is supported on the operating system level and without which our research would have been much more difficult.

We typically run *Media Streams* in an 80 Megabyte RAM partition on an Apple Macintosh Quadra 950 or Quadra 840AV with two high resolution (21 inch), accelerated 24-bit color displays. We use the SuperMac DigitalFilm board for digitization and recompress our QuickTime movies to Apple PhotoJPEG, and then recompress them again to Apple's CinePak format in order to reduce file size and enable software-only 30 fps playback under QuickTime 2.0 (believe it or not, this type of procedure is the recommended one in the QuickTime 2.0 documentation—the trouble

with inventing the future is that we have to live in the present). We use a 5.5 Gigabyte MicroNet Raven '040 disk array for storage.

3.2. Design History

I began working on Media Streams in the fall of 1990 and in the spring of 1991 Brian Williams and Golan Levin began working with me as UROPs (Undergraduate Research Opportunity Program students) at the MIT Media Lab. We have been working on Media Streams ever since and we will continue working together at Interval Research Corporation in Palo Alto, California.

As a design team we worked together on almost all aspects of system and interface design. Golan, Brian, and I have had numerous and lengthy design meetings over the past four years and have refined a design process in which software/signal-processing engineers (Brian) and graphic designers/musicians (Golan) and artificial intelligence researchers/media theorists (myself) can work together.

The result is a working system with 2.49 Megabytes (41,920 lines) of source code and over 3,500 icons in its visual language. Our current video database has 24.07 minutes of footage with 2,090 annotations. There are 17 different movies, of average length 1.4 minutes.

Media Streams has had three major versions. The version discussed in this thesis is 3.1. The transition from version 2 to version 3 saw the most major revision of the interface design.

In the remainder of this chapter we will briefly outline Media Streams' main functionality and system components so as to provide a frame for the more detailed discussion of its video representation, retrieval mechanisms, and user interface in the succeeding chapters. Throughout this chapter I will refer to interface components that are discussed in greater depth in Chapter 6 and are beautifully illustrated and explained in detail in the excellent *Media Streams 3.0 User's Guide Manual* (Levin 1994) which may be found in Appendix A.

3.3. Media Streams Functionality

Media Streams enables users to annotate, browse, retrieve, and repurpose digital video and audio content with an iconic visual language designed for

video representation. Its main functions are outlined in the following subsections.

3.3.1. Preprocessing

Media Streams makes use of existing and reliable signal-processing techniques for automatically creating meaningful segmentations and visualizations of digital video and audio data. When a QuickTime movie is first loaded into the system Media Streams creates scene-breaks for the video and pause-breaks for the audio. The system also automatically creates multiple representations of the video and audio data's structure at different temporal and spatial resolutions (for video: thumbnails and a videogram; for audio: waveforms and pause-break bars) which are used in visualizing and navigating the data.

3.3.2. Annotation

In Media Streams, annotators use an iconic visual language to create stream-based annotations of video content. Media Streams utilizes a hierarchically structured semantic space of iconic primitives which are combined to form compound descriptors which are then used to create multi-layered, temporally indexed annotations of video content. These iconic primitives are grouped into descriptive categories designed for video representation and are structured to deal with the special semantic and syntactic properties of video data. These categories include: space, time, weather, characters, objects, character actions, object actions, relative position, screen position, recording medium, cinematography, shot transitions, and subjective thoughts about the material.

In Media Streams, the annotation language is designed to support the annotation of the consensual aspects of video content—what one sees and hears, rather than what one infers from context—in order to facilitate the convergence of iconic annotations and the repurposability of the content described by these annotations. Media Streams does not aim to support all types of annotations, but only those physically-based descriptions whose semantics supports repurposing. Other types of annotations may be layered on top of and use those created in Media Streams, but the goal here is for finding the most minimalist way of saying the most salient things about the content so as to support content-based retrieval for repurposing.

Media Streams' annotations do not describe video clips, but are themselves temporal extents describing content within a video stream. As *stream-based* annotations they support multiple layers of overlapping descriptions which, unlike clip-based annotations, enable video to be dynamically resegmented at query time.

The system also supports the reuse of other people's descriptive effort through the ability to retrieve and group related iconic descriptors into palettes.

3.3.3. Browsing

Browsing in Media Streams can make use of the representations of video content which are automatically generated as well as the annotations created by human users. For example, users can use a jump button (identical to the "track advance" button on consumer CD players) in order to jump by content to the next logical change in the video stream be it the next scene break or the next new character in a scene.

3.3.4. Retrieval

Media Streams supports the retrieval of annotated video segments and sequences in two ways: *by description* or *by example*. Query by description is the use of the annotation language as a query language in order to describe footage that one wants to find. Query by example is using already annotated footage itself as a query. Unlike most conventional video retrieval systems, Media Streams supports query of annotated video according to its *temporal* and *semantic* structure.

3.3.5. Repurposing

Media Streams is designed towards supporting the repurposing of video content in all of its functions and components. The functionality that most clearly shows this is the way in which Media Streams redefines retrieval in terms of composition. A query for a video sequence will not only search the annotated video streams for a matching sequence but will *compose* a sequence out of parts from various videos in order to satisfy the query. We refer to this retrieval strategy as *retrieval-by-composition*. Even on the level of satisfying user queries, Media Streams can repurpose the content in its own archive in order to *make* video sequences as a way of satisfying requests to *find* them.

3.4. Media Streams Components

3.4.1. System Data Structures

The composable primitives used to describe video are organized into a FRAMER structure that is hierarchical, extensible, and *semantic*. We call these structures **CIDIS** (CIDIS stands for “cascading icon dialog items” which is the name of the interface components we had to invent in order to write and read these hierarchically structured, compositional primitives). To understand the semantic structure of CIDIS let us consider, for example, the CIDIS for “Jane” and “waves.” The CIDI for the character “Jane” is under the CIDI for “adult female” that is under the CIDI for “female,” etc.; the CIDI for “waves” is under the CIDI for “arm action” that is under the CIDI for “single body action.” Since CIDIS are not keywords, there is also a CIDI for “waves” that is under the CIDI for “natural aquatic object”. Thus “waves” the aquatic object is differentiated in the representation from the other CIDI for “waves” the body action. The user interface to the CIDIS is the Icon Space (Figure 5).

Media Time Lines are the FRAMER structures which represent *relational* and *temporal* structures between CIDIS. If we consider, for example, the CIDIS for the character “Jane” and the character action “waves,” a Media Time Line structure could represent the information specific to a given video segment in which “Jane is in the video from frame 100 to frame 200 and from frame 140 to frame 160 Jane waves.” In addition to the particular in and out frame information, the Media Time Line structure would express that the CIDIS are related to each other in case-frame-like relationships (“Jane” is the *subject* of “waves”) and in symbolic temporal relationships (“Jane” *contains* “waves”). The user interface for writing and reading these annotations, and for browsing and retrieving video with them is the Media Time Line (Figure 6).

A **compound-icon-index** keeps track of all the unique combinations of CIDIS and their occurrences on Media Time Lines. This structure is used to improve the efficiency of the system and is not visualized to the user.

Media Streams’ representational structures have dual lives: as internal system representations they exist as, are stored in and restored from, a FRAMER database (FRAMER’s objects are called “frames”). As interface components they are Macintosh Common Lisp CLOS objects. In Media Streams, CLOS objects point to the FRAMER frames they represent and vice-versa.

3.4.2. System User Interface Components

Media Streams attempts to address two fundamental interface issues in video annotation and retrieval: creating and searching the space of descriptors to be used in annotation and retrieval; and visualizing, annotating, browsing, and retrieving video shots and sequences. Consequently, the system has two main interface components: the Icon Space (Fig. 5) and the Media Time Line (Fig. 6).

The **Icon Space** is the interface for the selection and compounding of the iconic descriptors in Media Streams (Fig. 5). To date Media Streams has over 3500 iconic primitives. In the **Icon Workshop** portion of the Icon Space (the upper half) these primitives can be compounded together to form compound icons. Through compounding, the base set of primitives can produce millions of unique expressions. What enables the user to navigate and make use of a large number of primitives is the way the Icon Workshop organizes icons into cascading icon hierarchies. The Icon Workshop has two significant forms of organization for managing navigational and descriptive complexity: a cascading hierarchy with increasing specificity of primitives on subordinate levels; and compounding of hierarchically organized primitives across multiple axes of description.

In the **Icon Palette** portion of the Icon Space (the lower half of Figure 5), users can create palettes of iconic descriptors for use in annotation and search. By querying the space of descriptors, users can dynamically group related iconic descriptors on-the-fly. The Icon Palette enables users to reuse the descriptive effort of others. When annotating video, users can make use of related icons that other users have already created and used to annotate a similar piece of video.

The **Media Time Line** (Figure 6) is the core browser and viewer of Media Streams. It enables users to visualize video at multiple timescales simultaneously, to read and write multi-layered iconic annotations, and provides one consistent interface for annotating, browsing, and retrieving video and audio data. The categories for video annotation found at the top of the Icon Workshop are also found on the left hand side of the Media Time Line and designate the types of annotations users can make.



Figure 5: The Icon Space

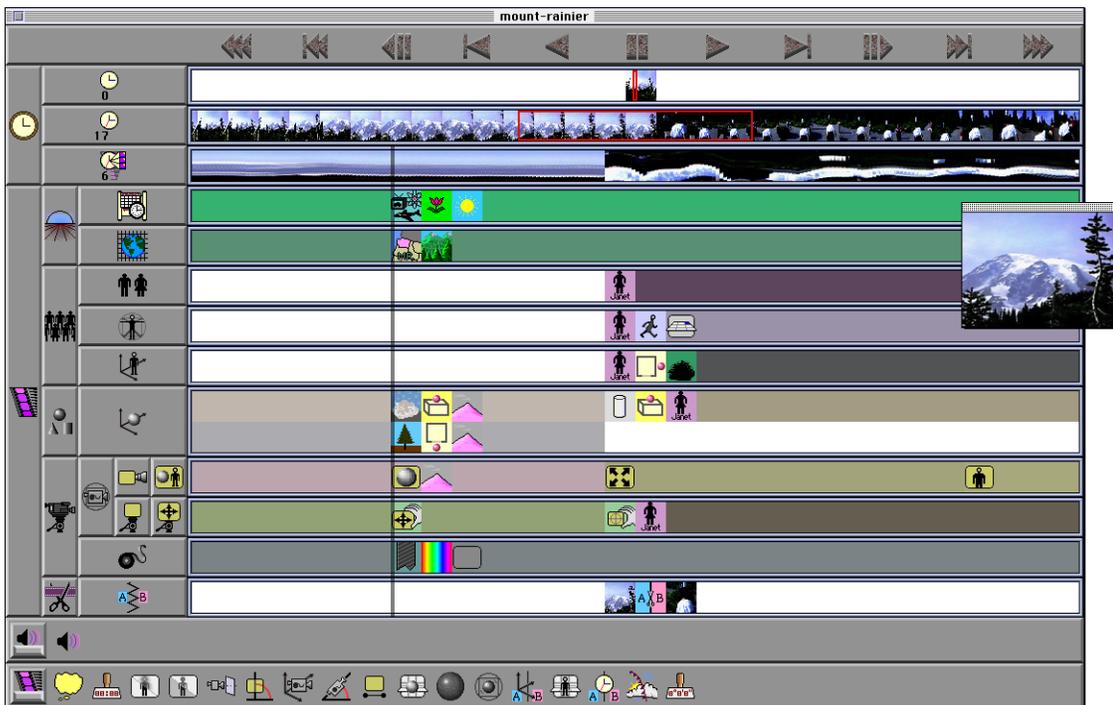


Figure 6: The Media Time Line

The process of annotating video in Media Streams using these components involves a few simple steps. In the Icon Space, the user can retrieve related iconic descriptors to form a customized icon palette or create iconic descriptors by cascading down hierarchies of icons in order to select or compound iconic descriptors. By dragging iconic descriptors from the Icon Space and dropping them onto a Media Time Line, the user annotates the temporal media represented in the Media Time Line. Once dropped onto a Media Time Line, an iconic description extends from its insertion point in the video stream to either a scene break or the end of the video stream. The flowchart below illustrates these steps and the dependencies in the process of annotating video in Media Streams (Levin 1994: 46).

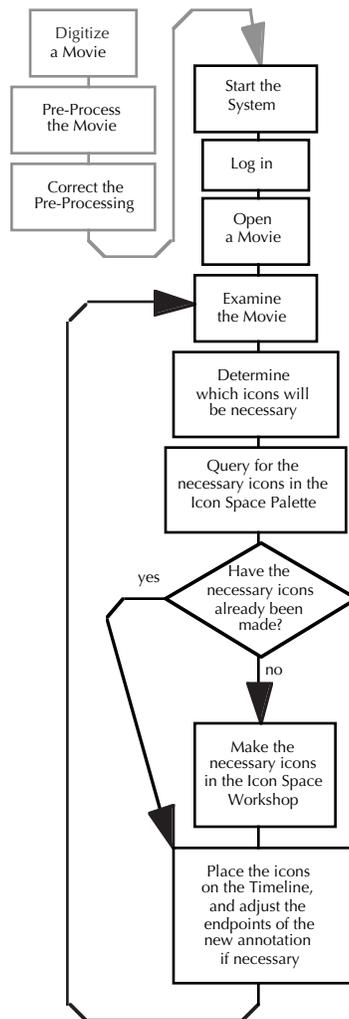


Figure 7. Typical Work Flow

By annotating various aspects of the video and audio stream (time, space, characters, characters' actions, camera motions, etc.), the user constructs a multi-layered, temporally indexed representation of video content.

In addition to dropping individual icons onto the Media Time Line, the user can construct compound icon sentences on the Media Time Line, which, when completed, are then available for use in the Icon Space and may themselves be used as descriptors. For example, the user initially builds up the compound icon sentence for "Jane waves" by successively

dropping the icons  and  onto the Media Time Line. The

user then has the "glommed" icon sentence  in the Icon Space to use in later annotation.

The interface for retrieval is the same as the interface for annotation: annotation is the process of describing footage one has; storyboarding is the process of describing footage one wants to make; query formulation is the process of describing footage one wants to find. In Media Streams, one interface is used for annotation and retrieval-by-composition.

3.5. Interaction Between Interface and Representation

In the design of Media Streams, the user interface and the underlying representations in the system were very closely linked: we used each to think through the other. Video is an inherently visual medium and we thought about it in visual terms. Developing the interface became a way to help us think through problems of representation and retrieval. However, separating the representation from the user interface is useful for the purpose of discussion in that it allows the respective strengths and weaknesses of each to be evaluated on their own. Therefore the next three chapters will discuss in turn the issues we faced and the solutions we developed for representing video content, retrieving and repurposing it, and the user interface and visual language for accomplishing these tasks.



Chapter Four

Representing Video

4. Representing Video



The core task of this thesis is the intellectual investigation and designed implementation of a representation of video that enables humans and machines to annotate, retrieve, and repurpose video according to its content.

This task involves the application and crossbreeding of methodologies and technologies from various disciplines, chief among them, knowledge representation and film theory. Our approach creates a hybrid methodology between the design and implementation centered approach of knowledge representation and the methodology begun in the early part of this century by the techne-centered (Soviet formalist and constructivist) film theorist-practitioners, who, trained as engineers, tried to understand the structure and function of the nascent medium of motion pictures.

In attempting to represent the content of video, our hybrid methodology benefits particularly from the strategies of approaches to the analysis of texts and films developed by the reader-response school of criticism and theory (Suleiman and Crosman 1980; Tompkins 1980). The chief exponents of reader-response theory are Wolfgang Iser for text (Iser 1974; Iser 1978; Iser 1989a; Iser 1989b; Iser 1993) and David Bordwell for film (Bordwell 1985; Bordwell 1989; Bordwell and others 1985; Bordwell and Thompson 1990). Not surprisingly, both Iser and Bordwell share common roots in the work of the early Russian formalists whose own methodology influenced and was influenced by the techne-centered film pioneers.

Reader-response theory breaks with the tradition of author-centered criticism whose goal in analysis is to discover the author's intended meaning hidden in the layers of the text or film. In contrast to this approach, reader-response theory does not understand meaning as an object that could be created by an author and then encoded in a text to be decoded by a reader. Meaning is a process that takes place in the interaction between text and reader. This notion is summed up well by Roland Barthes, a semiologist whose theories both influenced and extended reader-response theory: "a text's unity lies not in its origin but in its destination." (Barthes 1977: 48).

In reader-response theory, texts and films are complex artifacts which readers/viewers interact with to produce a variety of interpretations, meanings, uses, and experiences. The challenge of analysis is not to find the meaning in the text, but to elucidate how the structures of the text condition the process of readers' interactions with it. In sum, the goal is not to tell us *what* texts mean, but *how* they mean. In order to accomplish

this analysis, reader-response theory uses a methodology similar to the engineering inspired methodologies of knowledge representers and the techne-centered filmmakers of the 1920's: the approach to the analysis of representational systems is to divide the task into understanding the *structure* and *function* of the object of inquiry.

In dealing with representational systems created by humans, structure and function can be analytically separated for the purposes of clarifying those structures which are intersubjectively observable in the object and which underlie and condition the various experiences and interpretations of that object. In Iser's terminology, the intersubjectively observable structures are called *textual structures*. The subjectively experienced but textually and socially conditioned realizations of these structures are called *structured acts*. Structured acts are not the simple replication of textual structures but their concrete realizations in various individual forms.

The task of analysis is to uncover the textual structures that support various structured acts. Similarly, our task is to uncover those structures underlying the representational system of video which support various structured acts by viewers. Specifically, we are asking what structures in video can we describe which condition the repurposability of video into new sequences? In order to answer these questions by creating a representation for video content we can investigate the structure and function of video. We will describe video's structure: What is it made of? What are its parts? We will investigate its function: How does it work? What are its effects?

Unlike reader-response analysts, our task here is not just to describe how texts mean for the purposes of analysis and interpretation, but to create computational representations of structure and function for the purposes of retrieval and repurposing, i.e., for the purpose of construction.

In this chapter we will examine the structures and functions of video, describe the representations we have developed for them, and close with a discussion of the prospects for extracting these representations automatically.

4.1. The Structures of Video

4.1.1. Basic Structures

In designing a representation of video content we must think about the structure of what is being represented. Video is the electronic child of that turn of the century wonder: motion pictures. What separates motion pictures from all other preceding visual art forms is twofold: the ability to represent motion by (apparent) motion; and montage, the ability to take sequences of recorded events and recombine them into new sequences.

In terms of their materials, motion pictures are a composite medium. At the coarsest level they have always combined visual and auditory elements. Even in the so-called “silent era” motion pictures were accompanied by instrumental music and sound effects performed in real time (Hofmann 1970; London 1936; Walker 1979). If we want to analyze the structure of motion pictures, we can perform two major types of segmentation of their structure—one “horizontal” and one “vertical.”

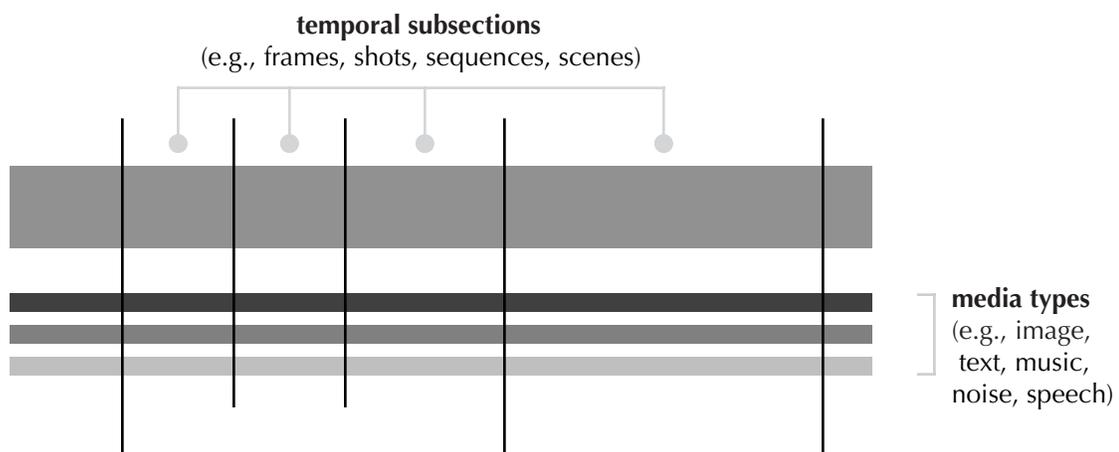


Figure 8. Horizontal and Vertical Segmentations of Video

The vertical segmentation divides video into streams of various media types. The horizontal segmentation divides these tracks into various temporal subsections which are discussed below.

4.1.1.1. Streams

Christian Metz segments video vertically into five separate streams of its media types (Stam and others 1992: 37). We can group these streams according to their auditory or visual modality and sub-group them according to their verbal or non-verbal content:

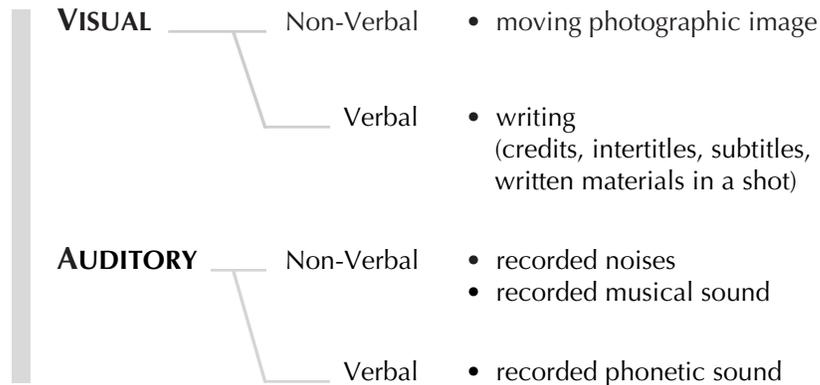


Figure 9. Vertical Segmentation of Video

These streams of media can be vertically segmented along the time axis. The most common temporal segmentations of video data are into frames, shots, sequences, and scenes.

4.1.1.2. Shot

Motion pictures create the illusion of motion by playing back a succession of still images at a rate faster than the rate at which we would look at them in a photo album and slower than the rate that would render incomprehensible those aspects of their content which remain continuous from image to image (Hochberg 1986). These still images are referred to as “frames” and the rate of their playback as the “frame rate” of a movie. A video camera produces a temporal stream of frames played back at a certain rate—normally 30 frames per second. The most basic segmentation of a stream of video frames is the *shot*. A shot can be understood as a stream of frames recorded between the time in which the recording device is turned on and turned off. The recording of a shot entails a continuity of time and space since the recording device will depict its continuous motion through time and space throughout the duration of its recording. A shot can be a single frame or hours upon hours of continuously recorded frames.

4.1.1.3. Sequence

The second innovation of motion pictures, montage, becomes most apparent when a movie is composed of two or more shots. According to Eisenstein's notion of montage, montage encompasses not only the concatenation of two shots, but can occur within a single shot through the juxtaposition of its individual elements (Eisenstein 1947; Eisenstein 1949; Eisenstein 1970; Eisenstein 1982; Eisenstein 1985). We will consider intra-shot montage as a metaphorical extension of the core notion of montage as the concatenation of two or more shots. A *sequence* is a montage of two or more shots. What makes cinema unique is the ways in which montage can juxtapose shots of discontinuous spaces and/or discontinuous times to compose new sequences of actions in spaces over time.

4.1.1.4. Scene

In traditional Hollywood cinema, a sequence of shots which appear to occur in the same *space*, over a continuous stretch of *time*, with an intelligible progression of *action* is referred to as a *scene* (Bordwell 1985: 158). A single shot can function as a scene though this is the exception rather than the rule in traditional cinema. Higher level vertical segmentations of video are of course possible. The task of most film analyses is to create a segmentation of a film at higher levels of organization than shots, sequences, or scenes. What makes these lower level segmentations so useful is that they are vertical segmentations of video which, on the one hand, can be intersubjectively established, and on the other hand, can form the basis on top of which higher level segmentations can be constructed.

In the area of digital signal processing of video many researchers have in recent years been developing "scene break" detectors (Arman and others 1992; Elliott 1993; Nagasaka and Tanaka 1992; Otsuji and others 1991; Tonomura and others 1993; Ueda and others 1991; Williams 1994; Zhang and others 1993). With few exceptions, these approaches have not developed a coherent framework, informed by the film theoretic analysis of video, for classifying the types of vertical segmentations in video (Hampapur and others 1994). One simple distinction that should inform video scene break detectors arises from the notion articulated here of the difference between a sequence and a scene: a scene break detector should have a model of the difference between *inter-scene* cuts (cuts between scenes) and *intra-scene* cuts (cuts within scenes). This is just one of a host of ways that a film theoretic understanding of the structure of video could

contribute to the development of more sophisticated video segmentation algorithms.

4.1.2. Representing Structures

At base video is a temporal medium that represents continuities and discontinuities of space, time, and action. The first task of a representation of video content is to provide a set of units into which the temporal streams of audio and video data can be parsed. In film theory, this task of dividing the streams of video and audio data into units is called *segmentation* (Bordwell and Thompson 1990: 49). The task of representing the basic structures of video data is the task of creating a useful segmentation of that data.

One might think that for the purposes of retrieval and repurposing a segmentation of video into frames, shots, sequences, and scenes would be sufficient. However necessary these segmentations are for video representation they are insufficient for representing video content. First of all, each of the vertical segmentations has certain inherent limitations as a content representation. Frames by themselves are too fine a segmentation and remove the temporal aspects of video content from a representation. Scenes are often too large of a segmentation to be useful for repurposing; by virtue of their completeness they render their parts less easily repurposable. Shots and sequences are a useful level of granularity but in and of themselves these segmentations do not represent their contents. Finally, and most importantly, there are many aspects of video content which *continue across shot and scene boundaries* (e.g., music, dialogue, character, etc.) or *exist within shot boundaries* (e.g., action, camera motion, etc.). These content streams offer additional vertical segmentations of video content.

Most current systems for representing and manipulating video create a segmentation of video into *clips*. As will be explained below, representing video by segmenting it into clips is a representational strategy that does not support multiple reuse of the representations or of the data represented. The core task of representing video for repurposing is to create a *segmentation of the data out of which multiple segmentations can be generated*. As will be explained below, a *stream-based* representation of video content enables multiple segmentations of video to be generated.

4.1.2.1. Clips vs. Streams

In most representations of video content, a stream of video frames is segmented into units called *clips* whose boundaries often, but do not

necessarily, coincide with shot, sequence, or scene boundaries. Current tools for annotating video content used in film production, television production, and multimedia, add descriptors (often keywords) to clips. There is a significant problem with this approach. By taking an incoming video stream, segmenting it into various clips, and then representing the content of those clips, a clip-based representation imposes a *fixed segmentation* on the content of the video stream. To illustrate this point, imagine a camera recording a sequence of 100 frames.



Figure 10. Stream of 100 Frames of Video

Traditionally, one or more parts of the stream of frames would be segmented into clips which would then be annotated by attaching descriptors. The clip is a fixed segmentation of the video stream that separates the video from its context of origin and encodes a particular chunking of the original data.

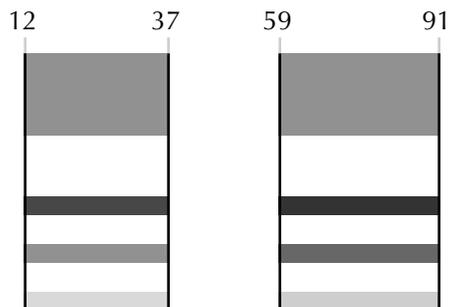


Figure 11. Two “clips” with Three Descriptors Each

In our representation, the stream of frames is left intact and is annotated by multi-layered annotations with precise time indexes (beginning and ending points in the video stream). Annotations could be made within any of the various categories for media annotation discussed below (e.g., characters, character actions, objects, spatial location, camera motions, dialogue, etc.) or contain any data the user may wish.

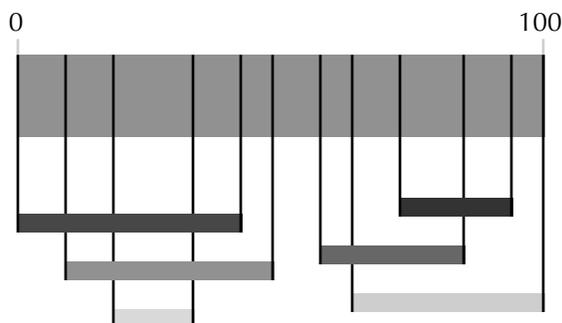


Figure 12. The Stream of 100 Frames of Video with 6 Annotations Resulting in 66 Possible Segmentations of the Stream (i.e., "clips")

In the above figure, only six unique annotations of a video stream yield 66 possible segmentations of the video stream. The formula used to calculate this result is:

$$S = {}_n C_2 = \frac{n(n-1)}{2}$$

S = number of unique segmentations

n = number of unique start-frames and end-frames

Stream-based representation makes annotation pay off—the richer the annotation, the more numerous the possible segmentations of the video stream. Clips change from being fixed segmentations of the video stream, to being the results of retrieval queries based on annotations of the video stream. In short, in addressing the challenges of representing video for large archives *what we need are representations which make clips, not representations of clips.*

A stream-based representation enables video data to be dynamically resegmented according to user queries. It makes annotation compositional and generative, such that the more a stream is annotated, the more possible segmentations it can yield. It also enables different levels of annotation to encapsulate each other such that, for example, annotations of scenes can make use of the annotations of shots which occur within them.

Today very few video systems use a stream-based representation due to the overwhelming influence of single-use models of video practice on the development of clip-based representations of video content. It is difficult for system designers to even think of alternate ways of segmenting video other than clips. There are a few other examples of stream-based representations of video content besides Media Streams. The work of Thomas Aguiere Smith (Aguiere-Smith 1992) together with

Glorianna Davenport (Davenport and others 1991) utilizes stream-based representations. Recently, Lee Morgenroth has begun to build on their work and the results look promising.

Structurally, video can be thought of as a multi-layered temporal stream of events of varying durations. A stream-based representation takes advantage of the structural properties of video in order to enable the task of representation to yield new segmentations. We now will discuss the functional properties of video and the representational strategies appropriate to them.

4.2. The Functions of Video

Attempts to represent video content from within the discipline of knowledge representation have largely ignored the fact that video is itself a representational system. Video data has been represented as if it were a transparent window on the world which one simply looks through to see the content to be represented. There are several unexamined and incorrect assumptions in this approach: that video images are a simple reproduction of the world, in semiotic terms “iconic” signs; that video has no unique semantics or syntax, either because it is a window on the physical world, which is not linguistic, or its content is represented by linguistic units which have the semantics and syntax of natural languages; and, consequently, that the understanding of video sequences requires no special types of cognition or knowledge.

In order to understand how video functions we can begin by understanding its properties as a representational system. We will attempt to examine and reorient the assumptions underlying artificial intelligence approaches to video representation by using ideas from film theoretic investigators of the representational functions of video, namely the semioticians Umberto Eco (Eco 1976a) and Christian Metz (Metz 1974).

In understanding the functions of video as a representational system we will be answering part of the reader-response question for video: *how* does it mean? In order to do this we will investigate the representational status of the video image, try to understand in what ways video is or is not a language; and discuss the particular semantic and syntactic properties of video sequences.

4.2.1. Representational Status of the Video Image

In designing a representation *of* video the first question one must confront is what kind of representation *is* video? Let us first limit our investigation to an individual frame, an image of video.

4.2.1.1. Some Basic Concepts: Saussurean Linguistics

From the earliest days of cinema, filmmakers and theorists have used linguistic tools to analyze motion pictures. They have sought to answer questions about cinema by comparing its structure and function to those of natural language. The linguistic theory that has had the most profound effect on film theory is *semiotics*. The founder of modern linguistics and the European grandfather of semiotics was Ferdinand de Saussure. Saussure taught linguistics in Geneva at the turn of the century and we know his work today through the course notes his students collected and published after his death (Saussure 1983).

Until Saussure, linguistics had largely concerned itself with languages and their history but not with language as such. Saussure shifted the focus of linguistics away from individual acts of speech (*parole*) to the study of the formal signifying system underlying language (*langue*). Similarly, Saussure's focus was less on the longitudinal history of languages over time (*diachrony*) than on the features of a language system (*langue*) at a given stage in its development (*synchrony*).

In analyzing *langue* synchronically Saussure arrived at what he believed were the fundamental characteristics of linguistic systems and their parts. He analyzed language's basic units and their basic principles of organization.

According to Saussure, the basic unit of language is the *sign*. Saussure's notion of the sign revolutionized linguistics and influenced much later thinking in film theory about the representational status of motion pictures. The sign is made up of two parts: the *signifier* and the *signified*. The signifier is the mental acoustic image, the inner sensory part of the sign; the signified is the mental abstract idea or concept that the signifier signifies. *The Course in General Linguistics* provides the following diagram and explanation of the signifier and the signified:

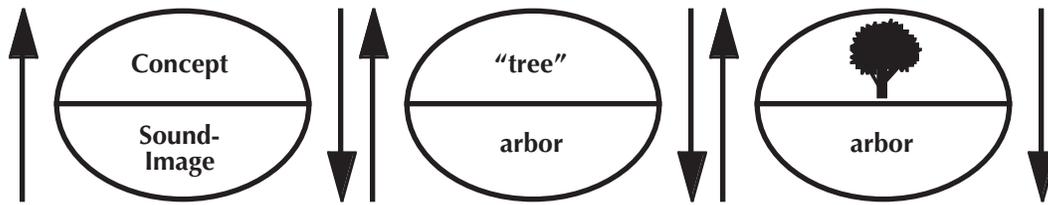


Figure 13.

The linguistic sign unites, not a thing and a name, but a concept and a sound-image. The latter is not the material sound, a purely physical thing, but the psychological imprint of the sound, the impression that it makes on our senses. The sound-image is sensory, and if I happen to call it "material," it is only in that sense, and by way of opposing it to the other term of the association, the concept, which is generally more abstract. (Saussure 1983: 66).

It is important to note that the signified is not the referent of the signifier. Signifiers do not signify things but ideas. This difference is crucial when we look at the signification of video signifiers which, though they may appear to resemble things in the world, do not stand for things but for concepts.

In contradistinction to much linguistic thought before him, Saussure made the *arbitrariness* of the linguistic sign one of its defining characteristics. For Saussure, it is the arbitrariness of the sign that enables signs to have a signifying as opposed to merely expressive function. If all signs were motivated we could not construct systems of signification (linguistic or non-linguistic). It is precisely the potential of any signifier to signify any signified that makes possible the synchronic diversity of sign systems, the diachronic development of sign systems, and the creation of higher order sign systems out of other sign systems (as in the sign system of myth as identified by Roland Barthes (Barthes 1972). What conditions the signification of signs is not their motivation (natural similarity to the signified) but the structures in which they occur. Saussure identifies two fundamental forms of organization for signs: *paradigmatic* (what he calls associative) and *syntagmatic*:

In discourse, on the one hand, words acquire relations based on the linear nature of language because they are chained together. [...] Combinations supported by linearity are *syntagms*. The syntagm is always composed of two or more consecutive units [...]. In the syntagm a term acquires its value only because it stands in opposition to everything that precedes or follows it, or to both.

Outside discourse, on the other hand, words acquire relations of a different kind. Those that have something in common are associated in memory, resulting groups are marked by diverse relations. [...]

We see that the co-ordinations formed outside discourse differ strikingly from those formed inside discourse. Those formed outside discourse are not supported by linearity. Their seat is in the brain; they are a part of the inner storehouse that makes up the language of each speaker. They are *associative relations*.

The syntagmatic relation is *in praesentia*. It is based on two or more terms that occur in an effective series. Against this, the associative relation unites terms *in absentia* in a potential mnemonic series. (Saussure 1983: 123).

The paradigmatic and syntagmatic are often conceived of as vertical and horizontal forms of organization. Syntagmatic structure is a horizontal sequence of signs in which the relation of the parts determines their meaning. Paradigmatic structure is a vertical space of substitutions in which a range of possible candidates can take the place of a given sign in the syntagmatic structure.

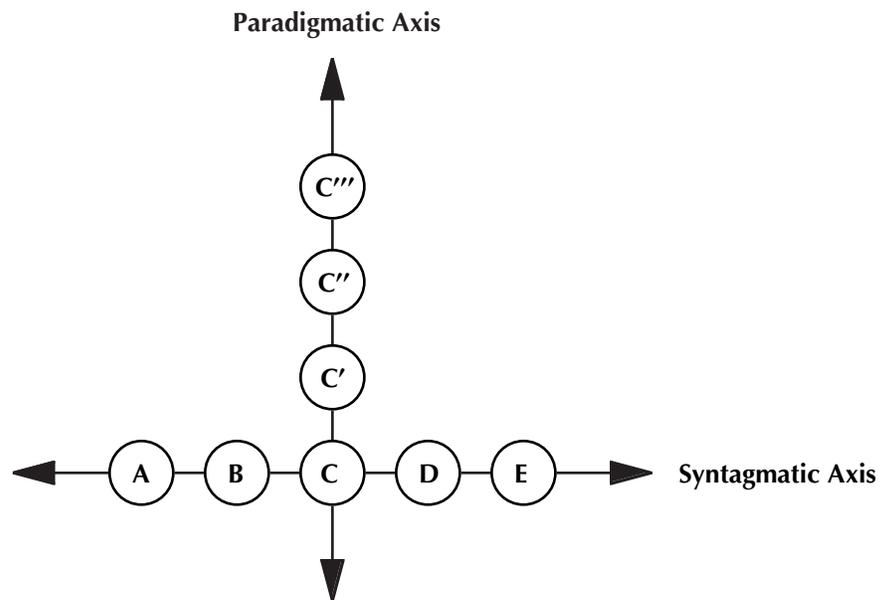


Figure 14. Paradigmatic and Syntagmatic Structures

Roman Jakobson extends the Saussurean notions of paradigmatic and syntagmatic organization by describing them as the more general processes of the *selection* and *combination* of signs (Hawkes 1977: 77-78). The ideas of paradigmatic and syntagmatic axes, of the vertical and horizontal structures of sign systems, provide useful tools for understanding the structures of many systems. As we will see in chapter five, the distinction in human memory between semantic and episodic memory structures could be understood as paradigmatic and syntagmatic organizations of remembered experiences.

Saussure understood syntagmatic and paradigmatic organization as not being an ordering of preexisting parts but a structuring that defines its parts by setting them into relation to one another. Saussure argues for the *diacritics* of sign systems. This aspect of Saussurean linguistics is often ignored by later semioticians who seek to understand signs as positive units out of which higher order structures are built rather than as differential units which are the results of these structures themselves. The striking exception to this trend is Jacques Derrida whose deconstruction of theories of language is deeply informed by Saussure's notion of diacritics (Derrida 1974). For Saussure, the signs of language are not positive units but only units of analysis which come into existence through differential relationships:

Everything that has been said up to this point boils down to this: in language there are only differences. Even more important: a difference generally implies positive terms between which the difference is set up; but in language there are only differences *without positive terms*. (Saussure 1983: 120).

The diacritical nature of sign systems, as we will see below, will be a defining issue in understanding the semiotic status of video.

Now that we have assembled a repertoire of conceptual tools from Saussurean linguistics (signs, arbitrariness, paradigmatic and syntagmatic structure, and diacritics) we can begin to unpack the question of the representational status of video.

4.2.1.2. The Video Sign: Motivated or Arbitrary?

If we understand a video frame as a sign, the question that divides many semioticians and film theorists is whether that sign is *arbitrary* or *motivated*. One might think that by placing a camera in the world and recording that of course the signs of cinema would be motivated, that one need only look in order to understand. Most people accept that the word *dog* is not the

same as a dog, but why do we think that an image of the dog is the same as a dog? Clearly the materials of the image are not the same. But one might say, that an image of a dog *looks* like a dog. However, seeing an image of dog is not the same as seeing a dog in the world. The act of representation itself, of framing, constructing, and eliciting a certain kind of seeing is a *cultural* practice that relies on many codes in order to be intelligible. The varieties of representational codes that humans have developed for representing reality attest to the constructed quality even of the “mechanical” reproductions of cinema. Nevertheless, the idea that cinema is made up of immediately intelligible images of reality is a belief common to an entire direction of film theory spawned by André Bazin (Bazin 1971). In Bazin’s school of thought, cinema’s task is to represent reality as closely as possible. Metz falls in the Bazinian camp, since his notion of the representational status of the cinematic sign is that the cinematic sign has a natural analogy to what it signifies and as such is motivated and not arbitrary. Metz does not consider, until driven to by Eco, that we must learn to read cinematic signs like all others and that underlying the cinematic sign is not a natural analogy to what it signifies but a set of cultural codes of perception, signification, and intelligibility.

Umberto Eco offers the most coherent and systematic critique of the naive notion of the natural motivated cinematic sign in his explication of the various codes of the cinematic image. For Eco, a single frame is situated within a myriad of perceptual and cognitive codes which enable us to recognize and understand its contents. Video is not a representation of the world, but a set of intelligible cues that we use to construct a mental image.

4.2.1.3. Cinematic Articulations: None or Three?

One of the defining characteristics of natural languages which Metz argues that cinema lacks is the quality of a *double articulation*. In natural language, the system of morphemes, the smallest units of significance, constitute the first articulation of language. The units of the first articulation (morphemes) are constructed out of the parts of the second articulation, the phonemes of language which in and of themselves are not meaningful. Metz argues that cinema has no second articulation, no base units out of which higher level meaningful units could be constructed through combination. Metz’s disbelief in articulations in the cinematic code stems from his conviction that the cinematic image is a natural analogon: as such it is indivisible into component parts and is not subject to articulation. In contrast, Eco identifies at least three articulations of the representational system of the cinematic image: photograms, iconic signs, and iconic semes:

Let's look again at the frame indicated by Pasolini—a teacher talking to students in a classroom. Consider it at the level of one of its photograms, isolated synchronically from the diachronic flux of moving images. Thus we have a syntagm whose component parts we can identify as semes combined together synchronically — semes such as 'a tall blond man stands here wearing a light suit ... etc.' They can be analyzed eventually into smaller iconic signs—'human nose', 'eye', 'square surface' etc., recognizable in the context of the semes, and carrying either denotative or connotative weight. In relation to a perceptive code, these signs could be analyzed further into visual figures: 'angles', 'light contrasts', 'curves', 'subject-background relationships'. (Eco 1976a: 601-602).

Eco's three articulations of the cinematic code provide an excellent framework for assessing the levels of a representation of video content. Current digital signal processing and computer vision algorithms strive to operate at the level of photograms (iconic figures) to identify the basic perceptual shapes and properties so of an image. What distinguishes our work is that we have formulated a representational language that operates at the level of iconic signs and iconic semes. A Media Streams log is a series of iconic semes made up of many iconic signs. We argue that this level of representation is the appropriate one for retrieval and repurposing, though it needs to make use of the photograms as well.

While Eco sees three articulations in the cinematic code of the image and Metz sees none, both agree that the granularity of the semiotic unit of a single video frame—the smallest possible video "shot"—is not analogous to a word, as many early film pioneers had asserted, but to a sentence.

4.2.1.4. Video Image: Word or Sentence?

In Metz's discussion of the shot-word analogy he makes no clear distinction between a shot and a frame. For him since the frame and shot have no articulations, they are fundamentally the same semiotically. For Eco that is not the case. He erects on top of his triple code of photograms, iconic signs, and iconic semes, which he developed for still images, another triple articulation of kinesic figures, kinesic signs (kines), and kinesic semes (kinemorphs) for describing the movement of bodies over time in motion pictures. For Eco, the level of complexity brought on in the transition from the frame to the shot only reinforces the arguments they both make about the disanalogy between the shot in motion pictures and the word in natural language. Metz sums up his objections to the shot-word analogy in the following five points:

It is true that the film sequence is a real unit—that is to say, a sort of coherent syntagma within which the “shot” react (semantically) to each other. This phenomenon recalls up to a certain point the manner in which words react to each other within a sentence, and that is why the first theoreticians of the cinema often spoke of the shot as a word, and the sequence as a sentence. But these were highly erroneous identifications, and one can easily list five radical differences between the filmic “shot” and the linguistic word:

- 1) Shots are infinite in number, contrary to words, but like statements, which can be formulated in verbal language.
- 2) Shots are the creations of the film-maker, unlike words (which pre-exist in lexicons), but similar to statements (which are in principle the invention of the speaker).
- 3) The shot presents the receiver with a quantity of undefined information, contrary to the word. From this point of view, the shot is not even equivalent to the sentence. Rather, it is like the complex statement of undefined length (how is one to describe a film shot completely by means of natural language?).
- 4) The shot is an actualized unit, a unit of discourse, an assertion, unlike the word (which is a purely virtual lexical unit), but like the statement, which always refers to reality or a reality (even when it is interrogative or jussive). The image of a house does not signify “house,” but “Here is a house”; the image contains a sort of index of actualization, by the mere fact that it occurs in a film.
- 5) Only to a small extent does a shot assume its meaning in paradigmatic contrast to the other shots that might have occurred at the same point along the filmic chain (since the other possible shots are infinite in number), whereas the word is always part of at least one more or less organized semantic field. (Metz 1974: 115-116).

The implications of Metz’s statements for the design of representations of video content are far-reaching and important. If shots are not analogous to words in their representational status, function, level, and complexity, how do we expect computer systems which use keywords (without the *relations* that would transform them into a syntagmatic structure) to describe video? Let us look more closely at Metz’s claims.

His first three claims are the least problematic and the most persuasive. Metz's first argument is about the space of possible shots. Shots are not limited in number as are the words in a lexicon, but are as numerous as the possible statements in a language. What Metz misses here is that by making this claim he opens up the possibility for Eco's triple articulations of the cinematic ode: if shots are analogous to sentences, what are the "words" that comprise them? For Eco, these would be the iconic signs which are themselves made up of photograms. It is possible that if Metz had given up on his naturalistic interpretation of the representational status of the video frame, he might have consented that motion pictures have articulations analogous to the double articulation of phonemes and morphemes found in natural language.

Metz's second argument that shots are not analogous to words is an extension of his first, in that the large number of possible shots does not already exist waiting to be used like words in a lexicon, but are like statements which first come into being when they are performed by actual speakers. As a side note we might consider how the semiotic status of the shot would change if no new shots were ever made (or available) and filmmakers only had access to a fixed pool of preexisting shots. It is likely that these shots would form a lexicon and function as words in a visual language. There is some odd empirical evidence for this in the story of a team of Antarctic scientists that appeared in the Wall Street Journal:

How boring is life in the Antarctic? People in one group wintering at the South Pole in the 1960s watched the film "Cat Ballou" 87 times. People in another, after tiring of the westerns, Disney features and pornographic films on hand, spliced the movies together into their own production and adopted a vocabulary based on their creation that was so strange that relief crews arriving in the spring could barely understand them. (Burrough 1985)

One might argue that in popular culture today we live in a larger version of the Antarctic where our conversations with one another repurpose a common pool of pop cultural materials. Metz's third claim would still enable us to see these common materials as being more analogous to sentences than to words even if they are used as "stock phrases". Metz argues that shots are so full of semantic content that they cannot be described by just one word. Most shots cannot even be represented by a sentence, but are at least a paragraph's worth of complex relationships between multiple parts. Metz also hints that the task of representing a shot by natural language per se may be flawed and we will revisit this point later in our discussions of action representation and Media Streams' iconic visual language.

Metz's fourth and fifth claims are more problematic in that they rely on his notion of the shot as an indivisible motivated representation of reality. His fourth claim is that a shot is an actualized unit of discourse, an assertion as opposed to a virtual lexical unit that can be used in assertions. One can reinterpret Metz's claim into something more persuasive and more interesting implied in his idea that a shot does not say "dog" but "this is a dog." A shot is not a word, but an *utterance* (Ducrot and Todorov 1979: 323-324). A shot is a piece of discourse, implicitly part of a narration, of an act of telling, even if the telling is the simple assertion of "this is an x." It is a representation that asserts that it is a representation. The shot of a dog is analogous to the utterance "this is a dog" in that it is an act of enunciation, of framing, not a virtual lexical unit, nor an unmediated image of reality. The filmic act of denotation is a self-referential act of deixis that implies a connotation of its representational activity. The cinematic sign functions as a sign whose sign making activity is part of its construction. Like Brechtian theater's "signification of signification" (Zeichen des Zeichens), cinema "advances pointing to its mask." The cinematic sign points to itself as an act of signification. It is thus not a transparent representation of reality, but an act of constructing representations which point to this act itself (Hecht 1977). Film shots are not words, but actualized utterances which connote the signifying practice of their denotation. Metz (after his concession to Eco's critique of the cinematic analogon) even seems to advance this point in his discussion of Jean Mitry's film semiology:

On the whole cinema is generally able to connote without needing *special* connotators because it has constantly at its disposal the most essential of all connotators—i.e. the choice between several ways of constructing the denotation. Inversely, it is because *the filmic denotation is itself constructed* (montage, framing, choice and arrangement of motifs), because it can never be reduced to any automatic functioning of iconic analogy, and because the film is not photography, that the cinema is able to connote without the constant help of separate signifiers of connotation. (Metz 1976: 575)

Metz's fifth claim is that shots in a sequence do not have the paradigmatic function of words in a sentence because the space of possible shots is so large. Metz is right and wrong in this claim. Shots do not have the paradigmatic function of words in a sentence, but they do have the paradigmatic function of sentences in a story. The range of possible shots is very large, but in actual sequences is fairly constrained by the codes of intelligibility of cinematic continuity, narration, and genre. Metz underestimated video's paradigmatic function because of his Bazinian view

of the representational status of the video image (though his discussion of connotation in Mitry's semiology of cinema would seem to controvert Metz's own fifth claim about the relative paradigmatic freedom of cinema). A coded construction always involves selection; the construction of a film image is a process of paradigmatic selection. In the case of movies made from parts of other movies, the paradigmatic aspects become especially important since so much of the composition is the act of selecting from a limited set of choices.

With the above qualifications and extensions, we have shown that a video shot is not analogous to a word, but to a multi-sentence utterance. Through our analysis of the representational status of video, we see the inadequacy of keywords as a representation of video content. Both Metz and Eco agree that a shot is not analogous to a "word" in cinematic language and cannot be represented by a word in natural language.

The remaining question to answer is how far this analogy to language can usefully be applied. The complete exploration of this question would be another thesis. We will complete our discussion by focusing on the semiotic investigation of the syntagmatic properties of video. Having established that video frames are non-motivated paragraph-like utterances that become intelligible by means of a triply articulated code, we can move from questions of the cinematic sign and the representational status of shots to questions of the syntagmatic and diacritical nature of cinematic language. In what way do sequences of video images function like syntagms in language?

4.2.2. Video and Language

Metz writes:

Going from one image to two images, is to go from image to language. (Metz 1974: 46)

For Metz, cinema is not a language system (langue) because it lacks the arbitrary sign, minimal units, and double articulation, but he considers cinema to function like a language (langage). Eco does not share Metz's views on the semiotic status of cinema and argues that cinema is not only like a language (langage) but has an underlying language-system (langue) as well. However, unlike Metz, Eco does not demand that for cinematic discourse to have an underlying code that its code be identical to the langue of natural language. In fact, as Saussure asserts in his prescription for a general semiology, Eco argues that all sign systems have underlying codes of which the langue of natural language is just one example. What distinguishes the codes of cinema from those of photography are the

syntagmatic possibilities of the two distinguishing features of cinema: motion and montage.

In moving from one image to two, we move from a discussion of the semiotic status of the image to an analysis of the syntagmatic properties of cinematic language. Whether *langue* or *langage*, cinema makes use of syntagmatic and paradigmatic forms of organization. The question that then arises is whether cinema is a diacritical system, like language, in which terms only take on their meaning in the presence of other terms (like words in a sentence), or a compositional system in which terms have meanings and take on new meanings in new contexts (like sentences in a story). Saussure writes of language:

Language is a system of interdependent terms in which the value of each term results solely from the simultaneous presence of the others [...]. (Saussure 1983: 114)

Apparently without knowledge of Saussure's work and decades before Metz or Eco, Eisenstein argues that cinematic language is a diacritical system as well:

The film-frame can never be an inflexible *letter of the alphabet*, but must always remain a multiple-meaning *ideogram*. And it can be read only in juxtaposition, just as an ideogram acquires its *significance, meaning*, and even *pronunciation* (occasionally in diametric opposition to one another) only when combined with a separately indicated reading or tiny meaning—an indicator for the exact reading—placed alongside the basic hieroglyph. (Eisenstein 1949: 65-66)

Eisenstein was on the right track though at the wrong level of representation. Shots are not letters or even words, but utterances whose semantics does radically depend on their position in a syntagmatic structure. The key insight of our representation for video is that a video shot has two entirely different semantics: *an invariant semantics independent of the sequence it occurs in and a variable semantics dependent on the sequence it occurs in*. In this sense, shots are positive units, which, when they are arranged in sequences, function diacritically. Taken out of context, video shots are actualized utterances with identifiable semantics. When these shots appear in sequences, their syntagmatic organization creates new levels of meaning which can completely override (as Eisenstein describes) and definitely inflect the semantics of the shots. The challenge for a video representation is to represent these two different semantics and the interactions between them.

4.2.3. Video Syntax and Semantics

In attempting to create a representation of video content, an understanding of the semantics and syntax of video information is a primary concern. For video, it is essential to clearly distinguish between context-dependent and context-independent semantics. Syntax, the sequencing of individual video shots, creates new semantics which may not be present in any of the individual shots and which may supersede or contravene their existing semantics. This is evidenced by a basic property of the medium that enables not only the repurposing of video data (the resequencing of video shots taken from their original contexts and used to different ends in new contexts), but motion pictures' basic syntagmatic functionality: the creation of meaningful sequences through the *montage* of visual and auditory representations of discontinuous times and discontinuous spaces.

4.2.3.1. The Kuleshov Effect and Montage

To determine the nature of montage is to solve the specific problem of cinema. (Eisenstein 1949: 48)

The early experimental evidence for the effects of the syntax of shot combination on the semantics of individual shots was established by the Soviet film pioneer Lev Kuleshov early in this century (Isenhour 1975; Kuleshov 1974). Kuleshov's work deeply influenced the Soviet montage school, all later Soviet cinema, and is a core inspiration for the research in this dissertation. Kuleshov is really the unsung hero of video representation. Kuleshov was himself an engineer who, after only having worked on one film, ended up helping found the first State Film School after the October Revolution. Lenin supported the early Soviet film industry because of his belief "that of all the arts for us the most important is cinema" because it was the most effective means for communicating the communist message to the proletariat (Lenin would have *loved* television) (Taylor and Christie 1988: 57). The one interesting hitch in the story of Kuleshov and early Soviet cinema is that for quite some time these filmmakers had no film stock (most of it having left the country with the exit of the heads of Russian film studios). In response, and well in keeping with the methodological tenor of the times, Kuleshov formed a workshop with his students and set about to make *films without film* both by "shooting" scenes with no film in the camera and by re-editing found footage.

Kuleshov was fascinated by the ability of cinema to create artificial spaces, characters, actions, and reactions through montage. Kuleshov sought to understand the compositional principles of montage which enabled viewers to create associations when viewing sequences of shots which if the shots were taken out of sequence would not be created. In his workshop, he and his students performed various “experiments” to investigate the space of effects made possible by montage—by the reordering of film shots into new sequences in such a way as to use them to create entirely new semantic effects (one might call Kuleshov’s experiments not only montage experiments, but repurposing experiments). Kuleshov recounts that all the materials and records of these early experiments are now lost and that many, many shot combinations were investigated. He finds that the “canonical” examples people write about are accurate in the theory they try to convey though not exact accounts of the experiments themselves (Kuleshov 1973: 70). The classic example of the “Kuleshov Effect” is recounted by Kuleshov’s student, Pudovkin, as the following experiment:

Kuleshov and I made an interesting experiment. We took from some film or other several close-ups of the well-known Russian actor Mosjukhin. We chose close-ups which were static and which did not express any feeling at all—quiet close-ups. We joined these close-ups, which were all similar with other bits of film in three different combinations. In the first combination the close-up of Mosjukhin was immediately followed by a shot of a plate of soup standing on a table. It was obvious and certain that Mosjukhin was looking at this soup. In the second combination the face of Mosjukhin was joined to shots showing a coffin in which lay a dead woman. In the third the close-up was followed by a shot of a little girl playing with a funny toy bear. When we showed the three combinations to an audience which had not been let into the secret the result was terrific. The public raved about the acting of the artist. They pointed out the heavy pensiveness of his mood over the forgotten soup, were touched and moved by the deep sorrow with which he looked on the dead woman, and admired the light, happy smile with which he surveyed the girl at play. But we knew that in all cases the face was exactly the same. (Pudovkin 1949: 140).

What the Kuleshov Effect reveals is that the semantics of video information is highly determined by what comes before and what comes after any given shot. In addition to creating artificial emotions through montage, Kuleshov experimented with creating artificial spaces and characters:

Yes; the experiments which followed those of the 'Kuleshov effect' are extremely interesting. They were concerned with 're-created space'; the action takes place in different places while the actors follow a single dramatic line, as if these quite separate places were adjacent to each other. [...]

What I think was much more interesting was the creation of a woman who had never existed. I did this experiment with my students. I shot a scene of a woman at her toilette: she did her hair, made up, put on her stockings and shoes and dress.... I filmed the face, the head, the hair, the hands, the legs, the feet of different women, but I edited them as if it was all one woman, and, thanks to montage, I succeeded in creating a woman who did not exist in reality, but only in cinema. (Kuleshov 1973: 70).

Kuleshov's experiments began the work of cataloging the effects and principles which underlie all montage and are especially important for a representation of video which seeks to repurpose content and retrieve sequences by composing segments from various videos. Because of the impact of the syntax of video sequences on the semantics of video shots, any indexing or representational scheme for video content needs to explain how the semantics of video changes through resequencing and recombination. The challenge for video representation is to provide a framework for determining and representing those aspects of video content whose semantics are invariant and sequence-independent and those whose semantics are variable and sequence-dependent.

What film theory teaches us is that a knowledge representation for video cannot rely on existing ontologies of the physical world. Video is itself a representational system with its own ontological properties and its own constraints on the construction and maintenance of representations of spaces, objects, and actions through the montage of shots. In a word, video has not only its own semantics and syntax, but its own "common sense" which previous approaches to common sense knowledge, temporal, and action representation have yet to address. In the following sections we will take the insights we have gathered from our discussion of the structure and function of video and describe the base ontology we have designed for representing the representational system of video.

4.3. Ontological Issues in Video Representation

4.3.1. Knowledge Representation for Video

Video representation requires the rethinking of traditional approaches to knowledge representation, retrieval, and generation in AI. Current attempts to represent video content utilize representations developed for other media. Most commercially used systems apply techniques used for representing text (predominantly keywords or full text annotation); AI-influenced representations apply techniques developed for representing the physical world (Guha 1994; Guha and Lenat 1994; Lenat and Guha 1990) or for representing abstract, supposedly media-independent concepts (Schank and Rieger 1985);(Schank 1993). All of these attempts neglect to consider that video as a data type may have unique properties which may themselves need to be explicitly represented and which may render techniques developed for other media inadequate. It is essential to understand that the “common sense” of video is not that of the world and that the simple application of common sense knowledge representations to video will be inadequate to the task of building systems for representing, retrieving, and repurposing video according to its content.

In current AI research on multimedia, the generation problem has been framed as the problem of constructing a media independent engine for creating sequences of concepts or events which then guide synthesis processes in different media: usually text (Meehan 1976; Schank and Riesbeck 1981); graphics (Feiner and McKeown 1990; Kahn 1979; Karp and Feiner 1990); and occasionally video (Schank 1993). With recorded video, the generation problem must be recast as a representation and retrieval problem. The task, as in editing together found footage, is a matter of creating media specific representations of video which facilitate the retrieval and resequencing of exiting content. This difference in approach has fundamental ramifications for representational design. It is not merely a matter of adapting media independent representations to the specific properties of video, but of designing representations whose basic ontology and retrieval mechanisms capture the specific semantic and syntactic properties of video.

Therefore, the task that confronts artificial intelligence researchers in this area is to gather insights from disciplines that have studied the structure and function of video data and to use these insights in the design of new representations for video which are adequate to the task of representing the medium. As we have illustrated, film analysis and theory have developed a useful repertoire of analytical strategies for describing the semantics and

syntax of video data. These insights inform the following theoretical discussion and representational design.

4.3.2. Types of Video

In designing a representation of video content we need to be clear about what is meant by “video.” The range of uses to which video technology is applied is incredibly broad and the subsequent types of video produced vary widely. Video may encompass footage of the inside of the human body taken for surgical exploration, abstract colors and shapes used in avant-garde video art, TV commercials, satellite data, home movies, pornography, news, training video, videos of meetings, videos of scientific experiments taken in time-lapse, soap operas—the list goes on and on. Rather than merely listing all types of video, we can create a classification space that will allow us to bring into relief the similarities and differences among various types of video content.

4.3.2.1. Abstract Classification of Video Types

We can begin to conceptualize the space of all video by identifying several important dimensions along which the various types of video can be classified and distinguished. The creation of these dimensions can enable us to compare various types of video so as to reveal the density or scarcity of individual types of video in various portions of the space as well as the suitability of different representations to different portions of the space of all videos. The potential dimensions of this space are restricted only by the requirements of the types of distinctions we would like to make within it.

Let us begin with the same set of assumptions we started with in thinking about video representation: cinema is about action. Specifically, what cinema is concerned with and what enables us to be concerned about it is that it depicts the temporally extended actions of human agents in physical spaces. Given this assertion, a few dimensions of the n-dimensional space of all video become clear: action, agent/patient, and setting. Finer distinctions can be made along each of these dimensions. For action, a significant distinction can be made between the types of actions the agents perform:

- **physical actions**
(motions of the body which can encompass a wide range of actions from locomotion to gestures)

- **vocal actions**
(actions performed by speaking agents which subdivide into verbal and non-verbal actions)

The types of agents and patients involved in the action are also especially significant in distinguishing different types of video. The conception of agency itself deserves a much longer discussion than can be conducted here, but a few important distinctions need to be made. In thinking about agents and patients, the concept of animacy comes to the fore: which agents and patients are animate, alive, like us? The notion of what constitutes animacy evolves throughout our lifetime and is often a fluid and context-dependent idea. Whether the agents or patients are animate greatly affects our understanding of the range of possible actions that can be performed by the agent on the patient. Animacy also affects our interest in the intentions and consequences of these actions. Animacy is not limited to human agents and patients. Animals, robots, even rolling rocks and billowing clouds can appear animate to humans given certain developmental, cosmological, or fictional stances (Ackermann 1991). In cinematic narrative, our attention is guided by the actions and attention of animate human agents who interact with other animate agents and the world around them. Video can be divided along dimensions of whether the agents or patients of actions are animate or inanimate, and if animate, whether human or non-human.

Most cinematic video takes place in settings which are recognizably human: artificially constructed interior and exterior spaces inhabited by humans on Earth within the time period of recorded human history. Settings outside of these dimensions articulate certain genres of cinema: science fiction, fantasy, nature documentary, and others.

Most of what we commonly think of as cinema inhabits a very particular part of this n-dimensional space of all video. We do not normally spend as much time watching video of grass growing or rocks eroding or humans suspended motionless in the ether, as we do watching humans interacting with each other, other animate agents and patients, and inanimate patients, in recognizable human settings.

Media Streams' representation is oriented toward this coherent, specific, yet large and significant part of the space of all videos: video of people and objects in human settings performing actions. Furthermore, I focus my investigation on short video sequences with a restricted range of actions: physical actions and non-verbal vocal actions. This is due to several factors:

- current technical limitations of video storage and playback
- the desire to solve a tractable problem that may be completed in a thesis-appropriate time frame (solving the problem of the representation of natural language content for video is itself worthy of several other doctoral theses)
- the desire to explore the representation and creation of a form of video that may be used in daily global, asynchronous, many-to-many communication

The short video sequence that is highly visually oriented (due to the absence of dialogue) and that makes use of non-verbal audio has a rich historical tradition and could prove to be an important form of communication between people around the world who do not have a common spoken or written language, but who may share a set of concerns and a common visual language.

4.3.2.2. Historical Precedents for Non-Verbal, Action-Centric Video

The historical sources that provide examples of this type of video may seem disparate, but in fact represent a common line of development through the history of the moving image. In the early days of cinema, before the consolidation of the Hollywood studio system, there was a period of wide ranging and vibrant experimentation in narrative and non-narrative forms of cinema (Elsaesser 1990). Most commonly this experimentation resulted in short film sequences accompanied by non-verbal audio (both music and sound effects). This period of experimentation bears a direct connection to the ongoing work of the avant-garde in the twentieth century. A related and important cinematic tradition is that of the compilation film that had and has its practitioners in news, documentary, and the avant-garde. The advent of music video in the late seventies and early eighties (MTV being the prime example) revived and borrowed visual traditions and styles from the early days of silent movies as well as the work of the avant-garde to create a new and increasingly global language of short image-centric videos. With the increasing availability of VCR technology, fans of television shows have created a unique genre of video that borrows from the conventions of music video, short narrative, the avant-garde, compilation films, and network television to create fan authored music videos which are entirely made out of found materials (Jenkins 1992).

These five historical sources—silent films, avant-garde films, compilation films, music videos, and fan films—form a rich source of stylistic, generical, and narrative forms useful in thinking about the look, feel, and uses of a future global language of video made possible by the availability of technology for annotating and retrieving video on a mass scale. The goal of this thesis work is to provide a representation language for the type of video common to all of these—video of people and objects in human settings performing actions—that will enable the development of new forms of videographic expression, communication, and use.

4.3.3. Base Categories for Video Representation

A central question in my research is the development of a minimal representation of video content. This has resulted in the development of a set of categories for, and a way of thinking about, describing video content. Let us build up these categories from examining the qualities of video as a medium. One of the principal things that makes video unique is that it is a temporal medium. Any language for annotating the content of video must have a way of describing temporal events—the actions of humans and objects in space over time. Therefore, we also need a way of talking about the characters and objects involved in actions as well as their setting, that is, the spatial location, temporal location, and weather/lighting conditions. The objects and characters involved in actions in particular settings also have significant positions in space relative to one another (beneath, above, inside, outside, etc.).

These categories—*actions, characters, objects, relative positions, locations, times, and weather*—would be nearly sufficient for talking about actions in the world, but video is a *recording* of actions in the world by a camera, and any representation of video content must address further video-specific properties. First, we need ways of talking about *cinematographic properties*, the movement and framing of the camera recording events in the world. We also need to describe the properties of the *recording medium* itself (film or video, color or black & white, graininess, etc.) Furthermore, in video, viewers see events depicted on screens, and therefore, in addition to relative positions in space, screen objects have *screen positions* in the two-dimensional grid of the frame and in the various layered vertical planes of the screen depth. Finally, video recordings of events can be manipulated as objects and rearranged. We create transitions in video in ways not possible in the real world. Therefore, *cinematic transitions* must also be represented in an annotation language for video content.

These categories need not be *sufficient* for media annotation (the range of potential things one can say is unbounded), but I purport they are *necessary* categories for media annotation in order to support retrieval and reuse of particular segments of video data from an annotated stream.

These minimal annotation categories attempt to represent information about media content that can function as a substrate:

- on top of which other annotations may be layered
- out of which new annotations may be inferred
- within which the differences between consensual and idiosyncratic annotations may be articulated

In *Media Streams*, the primary level of representation is of the semantically invariant, sequence-independent aspects of video content. The semantically variable, sequence-dependent aspects of video content are represented in terms of this primary level of representation. Therefore, the representational system is optimized to represent that which one sees and hears in a video shot, rather than what one infers from the syntagmatic context of a video shot. The process of representation is highly decontextualizing in order that these representations can support retrieval and repurposing of video content. The “Laws of Logging” at the end of the *Media Streams 3.0 User’s Guide and Manual* in Appendix A offer suggestions to users about how to log in this decontextualized way within our various categories of video representation.

Let us now expand our understanding of these categories (action, character, object, relative position, space, time, weather, cinematography, recording medium, screen position, and transitions) as an ontology for video representation.

4.3.3.1. Action

A core problem for representing temporal media is the representation of dynamic events. For video in particular, the challenge is to come up with techniques for representing and visualizing the complex structure of the actions of characters, objects, and cameras. A representation of cinematic action for video retrieval and repurposing needs to focus on the granularity, reusability, and semantics of its units.

There exists significant prior work in the formalization of temporal events in order to support inferencing about their interrelationships (Allen 1985) and to facilitate the compression and retrieval of image sequences by indexing temporal and spatial changes (Arndt and Chang 1989; Campanai

and others 1992; Del Bimbo and others 1992; Del Bimbo and others 1993). Media Streams creates a representation of cinematic action that these and other techniques could be usefully applied to. What these techniques do not consider is the representation of *cinematic* action with its unique semantic properties, granularities, and ability to be resequenced.

In video, actions have a dual semantics because their meaning can shift as the video is recut and inserted into new sequences (Isenhour 1975; Kuleshov 1974). For example, a shot of *two people shaking hands*, if positioned at the beginning of a sequence depicting a business meeting, could represent “greeting,” if positioned at the end, the same shot could represent “agreeing.” Video requires a representation of the variable and invariant semantics of action. In addition, the prospect of representing video for a global media archive brings forward an issue which traditional knowledge representation has largely ignored: cultural variance. The shot of two people shaking hands may signify greeting or agreeing in some cultures, but in others it does not. How are we to represent shots of people bowing, shaking hands, waving hello and good-bye? The list goes on.

In order to address the representational challenges of action in video, Media Streams does not represent actions according to their particular semantics in a given video sequence (a shot of two people shaking hands is not annotated as “greeting” or alternately as “agreeing”), but rather according to a *physically-based description* of the actions of bodies and objects in space. This physically-based description captures the invariant sequence-independent semantics of video.

In order to understand the level of representation at which our physically-based description needs to operate, we can turn to the categorization of the triple articulation of the codes of action in motion pictures that Eco builds on top of his triple articulation of the codes of the cinematic image.

4.3.3.1.1. Kines and Kinemorphs

Eco’s extension of his typology of the triple articulation of the codes of the cinematic image—iconic figures (photograms), iconic signs, and iconic semes—to a triple articulation of the codes of action in motion pictures—kinesic figures, kinesic signs (kines), and kinesic semes (kinemorphs)—provides the right levels of representation in order to differentiate the semantically invariant and variable components of cinematic action, though Eco does not himself make this realization. Eco does understand that action in cinema is not the same as action in the physical world and as such offers the possibility of being described as a code with multiple articulations:

Now kinesics has difficulty in identifying discrete units of time in the gestural continuum. *But not so with the camera.* The camera decomposes kinemorphs precisely into a number of discrete units which still on their own mean nothing, but which have different value with respect to other discrete units. If I subdivide two typical head gestures into a number of photograms (eg. the signs 'yes' and 'no'), I find various positions which I can't identify as kines 'yes' or 'no'. In fact, if my head is turned to the right, this could either be the figure of a kine 'yes' combined with the kine 'nodding to the person on the right' (in which case the kinemorph would be: I'm saying yes to the person on the right'), or the figure of a kine 'no' combined with the kine 'shaking the head' (which could have various connotations and in this case constitutes the kinemorph 'I'm saying no by shaking my head').

Thus the camera supplies us with meaningless kinesic figures which can be isolated within the synchronic field of the photogram, and can be combined with each other into kines (or kinesic signs) which in their turn generate kinemorphs (or kinesic semes, all-encompassing syntagms which can be added one to another without limit.) (Eco 1976a: 602-603).

In our representation, the semantics of kines are invariant and sequence-independent; the semantics of kinemorphs are variable and sequence-dependent. Media Streams' representation of action does not focus on the level of kinesic figures which are asemantic (though we do support their representation), but on the semantics of kines and kinemorphs. The representation of kines is discussed below. The representation of kinemorphs (and their sequence-dependent semantics of action) is accomplished through analogies between concrete instances of kines. Our implementation of the representational structures for kinemorphs is discussed in the next chapter on Media Streams' retrieval and repurposing mechanisms.

4.3.3.1.2. Representing Kines: Action Decomposition

In order to create a vocabulary for representing kines, the fundamental actions of bodies in space, our representation supports the hierarchical decomposition of its units both spatially and temporally.

Spatial decomposition is supported by a body-centered representation that hierarchically orders bodies and their parts which participate in an action. For example, in a complex action like driving an automobile, one uses one's entire body while the arms, head, eyes, and legs all function

independently. Actions of the hand may be thought of as subactions of the arm, eye actions a subaction of face actions, etc.

Temporal decomposition is enabled by a hierarchical organization of units such that longer sequences of action can be broken down into their temporal subabstractions all the way down to their atomic units. In the representational design of the CYC system, Lenat points out the need for more than a purely temporal representation of events that would include semantically relevant atomic units organized into various temporal patterns (repeated cycles, scripts, etc.). For example, the atomic unit of “walking” would be “taking a step” that repeats cyclically. An atomic unit of “opening a jar” would be “turning the lid” (which itself could theoretically be broken down into smaller units—but much of the challenge of representing action is knowing what levels of granularity are useful).

Direction of action is expressed in terms of the object toward which an action is oriented (e.g., “John walks *towards* Mary.”) and/or the screen position to which an action is directed (e.g., “John walks *screen right*.”).

There are also subcategories of action representation for actions particular to humans versus objects.



4.3.3.1.2.1. Character Actions

Human body motions are further represented in two ways: *conventionalized* physical motions and *abstract* physical motions.

4.3.3.1.2.1.1. Abstract Actions

The representation also provides a hierarchical decomposition of the possible motions of the human body according to articulations and rotations of joints. Since Media Streams enables multi-layered annotation, any pattern of human motion can be described with precision by layering temporally indexed descriptions of the motion of various human body parts.

4.3.3.1.2.1.2. Conventionalized Actions

There are, however, many commonly occurring, complex patterns of human motion which seem to have cross-cultural importance (e.g., walking, sitting, eating, talking, etc.). Conventionalized body motions compactly represent motions which may involve multiple abstract body motions.

One may ask “where is the representation of emotion in all of this?” If we remember the insights of the Kuleshov Effect, the answer becomes clear. Imagine a shot of a man *smiling*. Is it a “happy” shot? One might think so. But what if I edit this shot in a sequence so as to reveal that a gun is pointed at the head of the *smiling* man? Is he still “happy”? Perhaps the emotion is now better described as “fearful” or “pleading”? In both cases though the man is still *smiling*. Emotion is not a property of a shot that survives resequencing. Therefore Media Streams represents the underlying physiognomy of emotion by offering a typology of facial gestures, rather than emotions.

4.3.3.1.2.1.3. Object Actions



Object actions are subdivided into object *motions* and object *state changes*. Motions are inherently visual, like the action of a ball rolling or bouncing. State changes are only represented that are visually or acoustically perceptible, like the action of a ball burning.

4.3.3.2. Character



Identity of persons and objects is complex in video. A considerable portion of the cinematic craft is devoted to the construction and maintenance of coherent identities for characters and locales. This is achieved through the discipline of *continuity*. Continuity is the process whereby salient details of a character’s appearance remain in continuity from shot to shot (i.e., remain constant when appropriate, change when appropriate). For example, if an actor is wearing a black hat in one shot and not in the next, if there is no inferable explanation for the absence of the hat, then continuity is said to have been broken. The effort to maintain continuity is deeply related to the frame problem in AI (McCarthy 1958; McCarthy and Hayes 1969; Sack and Davis 1994). However, since video is not the physical world, but a systematic representation of it, continuity can be established and maintained by inferences not found in common sense reasoning.

There are two primary mechanisms for continuity of character, two ways the identity of a character can be established and maintained across a video sequence even if the shots come from different movies. These are *actor* and *role*. Actor refers to the actor’s body, to the visually and acoustically identifiable characteristics of a body which distinguish it from all others (like sex, age, body type, hair color/length, skin color, eye color, voice). Currently Media Streams supports the representation of *sex*, *age*, and *skin color* as defining body characteristics. Role refers to a complex of

expectations and cues which viewers use to identify a person as performing certain actions or having the ability to perform certain actions within appropriate settings. The main indicators of role are *costume*, *action*, and *setting*.

4.3.3.2.1. Actor

Continuity of character can cut across roles and be established solely by the continuity of actor. Shots of the same actor taken from various performances of different characters can be edited together to form one character. Imagine, for example, a story about a Kindergarten teacher who becomes a killer cyborg and then terrorizes a Martian colony. This sequence could be created by editing together several of Arnold Schwarzenegger's films (*Kindergarten Cop*, *The Terminator*, *Terminator 2: Judgment Day*, and *Total Recall*). Examples from popular television could make use of the crossovers of actor between television series like *Star Trek: The Next Generation* and *L. A. Law*. Imagine a sequence in which Dr. Pulaski (Diana Muldaur) is transported through time to stop Vash (Jennifer Hetrick) from marrying a member of the Q continuum who is masquerading as a divorce lawyer in Los Angeles (Corbin Bensen).

Continuity of actor can encompass different body parts making up one character if the various parts share or do not violate the characteristics of a unified body. Two examples are illustrative here. The first is the Kuleshov experiment of constructing one woman from shots of parts of different women; the other is from the *Archive Films Demo Reel* in which a sequence of a cab driver honking his horn is created by a shot of a cab driver followed by a close-up of a woman's hand honking a car horn (Archive Films 1992):

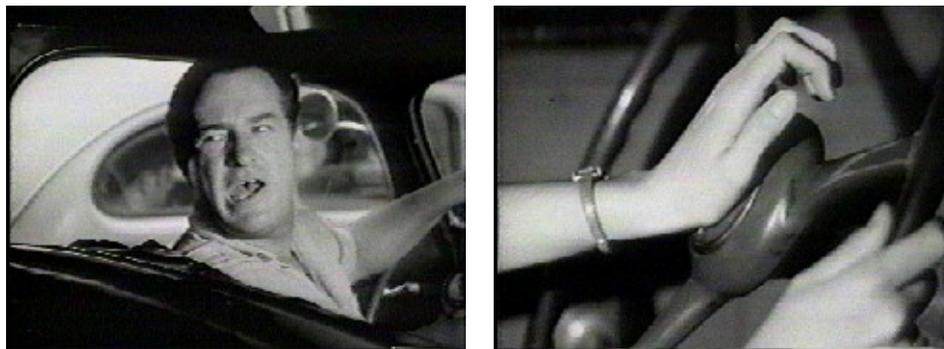


Figure 15. Constructed Continuity of Actor: A Kuleshov Effect

In this example, continuity of body is created by the lack of negative body cues (the arm is only recognizable as a woman's on close inspection) as

well as the continuity of action (body turning to honk horn, honking horn) and setting (inside a motor vehicle).

4.3.3.2.2. Role

Characters do not have “essential” identities in cinema. Characters are what they appear to be. For our purposes, someone dressed like a doctor *is* a doctor. Marcus Welby is an MD. Costume is the most significant determinant of role. The meaning of a costume (like a police uniform) can be inflected by action and setting. For example, a character dressed as a police officer driving a police car has a different role than a character dressed as a police officer robbing a bank. These are clearly different roles. The progression of narrative action may establish that these are the same character, but that would be a sequence level transformation of shot level indications of role. In Media Streams, we represent a character’s role by costume. Action and setting are concurrently represented by the categories of character action and mise-en-scene.

An example of continuity of character established by role that does not rely on continuity of actor also appears in the *Archive Films Demo Reel*, in which scenes of several different actors are cut together to make up the central character of a business man on his workday (Archive Films 1992):



Figure 16. Continuity of role in an assembled sequence

In this example, continuity of role is created by the continuity of costume (business suit), of action (go to the office, have a preliminary briefing, make a business presentation, go home to loving wife), and of voice (the first person narration).



4.3.3.3. Object

Objects have a bipartite representation, analogous to actor and role for character, which creates continuity of object: *form* and *function*. The formal properties of an object in video do not include its material, since it is the appearance which matters. In a sense, to video the world is a movie set and all objects are props. Media Streams' current object representation interleaves form and function. Later revisions to the representation will separate out these two properties.



4.3.3.4. Relative Position

The geometry of video spaces and the objects within them also have unique properties. The location of objects within the video frame can be represented by a hybrid 2-dimensional and 3-dimensional representation. Since video spaces can be constructed and concatenated into irreal geometries they have only a relational 3-dimensionality in which the geometry is best expressed in terms of *relative* as opposed to *absolute* positions.

Therefore, 3 dimensional spatial relations are represented by symbolic terms like "in front of", or "on top of", etc., instead of by an XYZ coordinate. Since the 3 dimensional world of video is itself represented in a 2 dimensional projection, all objects in the 3 dimensional space of the recorded/constructed world also have a corresponding location in the 2 dimensional plane of the screen.



4.3.3.5. Screen Position

The 2 dimensional screen position of an object is a crucial aspect of its spatial representation that is used by filmmakers to create both aesthetic order (in terms of balanced compositions as in photography) and cognitive order (in terms of the "rules" of Western filmmaking for the construction of space through action, chief among them being the "180 degree rule" that results in the well-known shot reverse shot of two person dialogue crosscutting).

4.3.3.6. **Mise-En-Scene: Space, Time, and Weather**

4.3.3.6.1. **Space**



Through the sequencing of shots, video enables the construction of many types of spaces: representations of spaces which have real world correlates (real spaces); spaces which do not but could exist in the physical world (artificial spaces); and even spaces which cannot exist in the physical world as we commonly experience it (impossible spaces). In thinking about the first two classes of spaces which can be constructed cinematically (real and artificial spaces) an important distinction can be made among three types of spatial locations:

- the actual spatial location of the recording of the video
- the spatial location which the viewer of the video infers when the video is viewed independently of any other shots
- the spatial location which the viewer of the video infers when it is viewed in a given sequence

For example, imagine a shot filmed in a dark alley in Paris on October 22, 1983, from 4:15 AM to 4:17 AM. The actual location of recording may be in a given street in a certain part of the city and could be expressed in terms of an exact longitude, latitude, and altitude. The shot we are imagining has no distinguishing features which mark it as a particular Parisian street or as a Parisian street at all. Independent of any sequence it appears as a “generic dark alley in a city.” With the use of a preceding establishing shot, for example an aerial view of New York City at night, the shot now has the inferable spatial location of “a dark alley in New York City.” Therefore, representations of the spatial location of a video must represent the difference between a video’s actual recorded spatial location and its visually inferable ones. The actual recorded spatial location for this dark alley shot differs from its visually inferable spatial location. This distinction is vital to any representation for reusable archives of video data, because it captures both the scope within which a piece of video can be reused and the representability of a piece of video, i.e., some shots are more representative of their actual recorded spatial location than others.

Another crucial aspect of representing spatial location is the difference between interior and exterior spatial locations. If a spatial location occurs inside a windowless structure, its inferable spatial location can be made to be almost anywhere by the use of an exterior establishing shot. Videotape

a shot of me in my office, place an exterior shot of the CIA before it, and whammo, I am working away on video representation in the CIA. The extent to which this interior-exterior distinction can be used is again illustrated by stills from a sequence from the *Archive Films Demo Reel* (Archive Films 1992):



Figure 17. An establishing-shot sequence

This plane ride is really taking place on a bus, though the sequence of exterior plane shot to inside a windowless bus enables the inferable location of the bus shot to be translated vertically by 30,000 feet!



4.3.3.6.2. Time

The representation of time in video requires the same distinction made for representing space: the difference between actual recorded time and the two types of visually inferable time: the time that the viewer of the video infers when the video is viewed independently of any other shots; and the time that the viewer of the video infers when it is viewed in a given sequence. A further important distinction in narrative video must be made among three different types of temporal duration (Bordwell and Thompson 1990):

- *story duration* (the implied duration of the events of the entire story as opposed to the particular story events selected for presentation in the video)
- *plot duration* (the implied duration of the particular events presented in the video)
- *screen duration* (the actual duration of the video as screened)

Media Streams focuses on *screen duration* in its representation of the temporal extents of annotations. Some of the articulations of temporal transitions which exist on the level of *plot duration* are addressed in our representation of *cinematic transitions*. However, the complete representation of these three types of temporal duration is an open research problem.

4.3.3.6.3. Weather



Weather is the final aspect of setting in our representation. Unlike weather in the physical world, weather in video is a visual and auditory as opposed to tactile experience. Therefore, weather is represented by properties of the shot which can be seen and/or heard: *moisture* (clear, partly sunny, partly cloudy, overcast, rainy, and snowy) and *wind* (no wind, slight wind, moderate wind, and heavy wind). Temperature is not something that can be directly seen: a video of a cold clear day may look exactly like a video of a hot clear day. It is the presence of snow or ice that indirectly indicates the temperature. “Weather” is a representation of the lighting and atmospheric conditions of outdoor shots. The representation of “indoor weather”—namely, lighting—is a necessary adjunct to our representation that remains to be done in the future.

4.3.3.7. Cinematography



Through discussion with people who have everyday experience with Hollywood-style film production and by researching camera description languages in film theory (Bordwell and Thompson 1990), we have developed a camera language that is both comprehensive and precise. In order to represent the cinematographic aspects of video we conceptualize the motion of the recording device that produced the images which the annotator sees. In cinema, the recording device typically has three interconnected parts which move independently to produce complex camera motions. The lens of the camera moves (to create different framings, zooms, etc.), what the camera is on—either a tripod or someone’s hand—moves (to create pans, to track a moving figure), and what the camera support is on—a “truck” or “dolly” in cinematic terms, or someone’s legs, or even a vehicle as in the case of shots taken from a moving car—may move as well (to create truck in, truck out, etc.). Each part of the recording device may also have important states as in the focus, camera angle, camera height, etc. In Media Streams, camera motions are described by multi-layered temporal descriptions of the actions of these camera parts: “lens” actions (framing, focus, exposure), “tripod” actions (angle, canting, motion), and “truck” actions (height and motion).



4.3.3.8. Recording Medium

In addition to representing the motions and states of the recording device we also represent the “look” of the recording medium. We represent the stock (70 mm film, 8mm video, etc.), color quality (color, black & white, sepia, etc.), and graininess (fine, medium, coarse, etc.) of the recording medium.



4.3.3.9. Transitions

The categories described above enable the user to produce representations of the content of video at the shot level. Transitions between shots are both the tools editors use to construct scenes and sequences out of a series of shots, and the gaps in a video stream of recorded space-time which are bridged by the viewer's inferential activity (Bordwell 1985; Bordwell and Thompson 1990). For example, if a viewer sees the two shot sequence:



Shot 1: person enters elevator,
elevator doors close



Shot 2: elevator doors open,
person exits elevator

Figure 18. A Two-Shot Elevator Sequence

The viewer infers that a certain amount of time has passed (probably between 10 and 60 seconds) and that a certain type of spatial translation has occurred (motion either up or down within a structure). Noël Burch has developed a systematic categorization of spatio-temporal transitions between shots in cinema (Burch 1969). He divides temporal transitions into continuous, forward ellipses in time of a determinate length, forward ellipses of an indeterminate length, and the corresponding transitions in which there is a temporal reversal. Spatial transitions are divided into continuous, transitions in which spatial proximity is determinate, and transitions in which spatial proximity is indeterminate. Burch's categorization scheme was used by Gilles Bloch in his groundbreaking work in the automatic construction of cinematic narratives (Bloch 1987). We adopt and extend Burch's categorization of shot transitions by adding

“temporal overlaps” as a type of temporal transition and the category of “visual transitions” for describing transition effects which unlike traditional cuts, can themselves have a duration.

Our work on a representation for cinematic transitions is just the first step towards a more complete representational scheme for cinematic transitions. Many refinements and extensions are possible. On the one hand, there are far greater levels of detail possible within the temporal and spatial transitions which Burch suggests. If a shot appears to be a “forward ellipsis in time of a determinate length” there is a multitude of useful subcategorizations of the amount of time that has appeared to transpire (seconds, minutes, days, etc.). Similarly for spatial transitions, the notion of determinate spatial proximity can be further detailed and clarified by a host of spatial transitions which regularly occur. The ways in which viewers build up models of cinematic space and the development of ways of representing this is an open and important area of research. Finally, there are a host of more complex and important syntagmatic cinematic structures which Burch’s categorization does not address but which Metz does in some detail. In fact, Metz’s major contributions to the theory of film and its relation to language are his classifications of the syntagmatic categories of cinematic discourse. Further research in the representation of cinematic transitions would have to begin with a thorough reassessment of Metz’s classification scheme.

Though our extended Burchian model is limited, it does allow our representation to articulate some basic transitions in time and space which serve to enrich the system’s representation of the syntagmatic functions of video.

4.3.4. Compositional Hierarchical Semantics

The above base categories form the top levels of our ontology for video representation. Terms in these ontologies are organized in semantic hierarchies from more general terms to more specific terms. In Media Streams, these levels in our hierarchy are represented as levels of FRAMER annotations which are linked in prototype relations. The relation between an upper level term and its subordinate terms can express a variety of relationships. The relations which our prototype relations capture the most often are:

- **class/instance**
(e.g., adult female/Tori Amos)
- **class/subclass**
(e.g., entertainer/singer)

- **whole/part**
(e.g., stereo system/speakers)
- **term/co-occurring term**
(e.g., toothpaste/toothbrush)

Media Streams' semantic hierarchies have two methods of combining primitive terms which enable our structured set of 3500 primitives to produce millions of valid compound expressions. The first method of combination occurs between *terms from the same semantic hierarchy* for a given category. Within many of Media Streams' base categories there are several subcategories whose terms can be cross-combined resulting in compound terms. For example, a spatial location has three top level subcategories: geographical location, functional location, and topological location. A compound term can be formed by compounding primitives from each of these subcategories—"Morocco", "mosque" and "inside"—into a compound spatial location—"inside a mosque in Morocco"—which itself can be used as a primitive term.

The second method of combination occurs between *terms from different semantic hierarchies*. This is a form of compounding which we refer to as "glomming" and it resembles the creation of case frames among terms. For example, we can create a glommed term between terms from characters (John), character actions (eats), and objects (pizza), to create the glommed term "John eats pizza." Of course, compound terms can be glommed as well. For example, we can create a glommed term with a compound term from characters (three adult female singers), a primitive term from relative position (in front of), and a compound term from objects (a blue piano), to create the glommed term "Three adult female singers are in front of a blue piano."

The representational design, semantic structure, and means of combination of the terms in our base ontology for video overcome the limitations of keyword-based or free text description. Media Streams offers a semantic representation designed for video content with an expressive and tractable combinatorics and semantics.

The question that remains to be answered is how much of video representation can be accomplished automatically. What we will find is that very little of the semantic level of representation discussed in this section can be derived automatically, but that current video parsing tools do create useful segmentations of video data which can provide important aids to humans using computers to annotate video content.

4.4. Automatic Representation

Current efforts in parsing video and audio signals in order to articulate their structures and contents do not even begin to describe the type of semantic level content discussed above. The automatic derivation of mappings between pixels and even mid-level perceptual entities such as people and objects is in its very nascent stages; the possibility of automatic parsing of high level semantic and cinematic categories is not even remotely achievable using current technology. Nevertheless, what current technology for automatically parsing video and audio is able to offer is significant for two reasons:

- Automated techniques can process very large corpora that humans may not have the time or interest to fully annotate.
- Automated techniques can greatly enhance the human annotation of media content through the design of “human in the loop” algorithms and systems.

An in-depth discussion of the various approaches and algorithms for video and audio parsing would fill another dissertation. For a preliminary survey of useful current approaches and a framework that integrates many of them into a working system, Brian Williams’ recent Bachelor’s thesis is a good place to start (Williams 1994). In the following sections we will briefly outline the current state of the art in automatic parsing of video and audio data, highlight the strengths and weaknesses of various approaches, and discuss the possibilities for advances in the next few years.

4.4.1. Video

Digitized video transforms an analog signal into a succession of XY matrices of pixels each representing its color value by a certain number of bits per color channel. This succession of XY pixel matrices can also be conceived of as an XY-T coordinate system in which spatial-temporal volumes can be represented. Most algorithms for video parsing have dealt with video as a succession of 2-D XY pixel matrices. Certain newer techniques are beginning to parse video as a 3-D XY-T pixel volume. Whichever representation is used for digital video, the capabilities of automatic parsing techniques fall into several broad classes of operations.

- Segmenting video sequences according to shot boundaries
- Recognizing camera motions
- Recognizing object motions
- Recognizing objects

Signal-based video parsing algorithms compute a series of low-level features from which these higher-level features can be derived. We will first describe the various low level features which are derived from computations on pixels and then in the following section describe the ways these low-level features are used to derive the higher level features mentioned above.

4.4.1.1. Computable Low-Level Features

In computing low-level features, video parsing algorithms look at pixels. They calculate two basic properties of these pixels from which all other properties are derived: *color* and *position*. Color is represented in various color spaces (RGB, YIQ, etc.). Most algorithms don't use the complete color information and reduce a color value to an *intensity* value that is obtained by the reduction of color values to gray scale in order to calculate an intensity based on gray level (e.g., in a 256 level gray scale intensity values range from 0 to 255). Position is expressed in terms of XY coordinates. What most analyses do is calculate values for the pixels of one frame and compare these to values calculated for the pixels of the succeeding frame. With these basic features simple analyses can be achieved and more complex analyses can be performed on more complex features derived from these basic ones.

4.4.1.1.1. Color Features

Computations that look at the color value of pixels either compare values for all pixels in a frame, which is inefficient if not implemented using parallelism, or more typically, calculate average values for regions of pixels in a frame. The earliest approaches to computing color features calculated *average color values* for regions in a frame (Sasnett 1986). Because these regions preserved the spatial location of pixels in the frame, comparisons between average color values also represented, and as a result were sensitive to, changes in camera or object motion. Techniques were then developed which overcame this sensitivity to camera and object motion by calculating a *histogram of color values* (Otsuji and others 1991).

Histogramming techniques represent the color values of pixels in regions of a frame as frequencies of occurrences of color values thereby throwing out the spatial coherence of the color data.

4.4.1.1.2. Motion Features

When a camera moves and/or what the camera is filming moves, the positions of pixels change from frame to frame. Some of the pixels can move, all of them can move, and they can move in any multitude of directions. It is important to realize too that a video camera represents the 3-D motions of the camera and/or object in a 2-D frame. By and large, without stereoscopic cameras, video provides insufficient information for recovering a complete representation of object and/or camera motion. Nevertheless, there are several types of motion represented the 2-D plane of the frame, and in the 3-D XY-T volume, that are recoverable and useful for video representation. The most basic techniques for comparing pixel motions are to simply compare the difference in positions between all of the pixels in two successive frames. This technique is so coarse as to be almost useless except as an aid to shot boundary detection algorithms. For recognizing camera and object motions a more sophisticated feature extraction process is used to approximate the vector of motion of pixels between frames. The most common method used is to compute *optical flow* (Horn and Schmuck 1981).

4.4.1.1.3. Compression Features

Recent techniques have attempted to avoid the computational load of calculating histograms and optical flow by operating on representations of video data in its compressed form. Some researchers have used the discrete cosine transform (DCT) coefficients from JPEG compressed video data as a feature representation for video frames (Arman and others 1992). In the cases where this method achieves inconclusive results, it is augmented by more computationally expensive color histogramming. Other researchers have looked at using the compressed frame size of video frames as a representation of its content (Deardorff and others 1994). Both of these techniques use very minimal but computationally cheap representations of the content of the video frame.

4.4.1.1.4. Static Structural Features

Recent advances in video parsing by the Vision and Modeling Group at the MIT Media Lab have centered on developing ways of capturing the structure of objects and their geometrical relations in a video frame. A technique known as *semantics preserving compression* uses an Eigenvector representation of video data to group structurally coherent parts of the frame (Pentland and others 1993b; Pentland and others 1994a).

4.4.1.1.5. Dynamic Structural Features

The attempt to capture static structure has been extended to techniques which can derive dynamic structural primitives by parsing coherent sections and patterns in the 3-D XY-T space-time volume of pixels (Bobick 1993).

4.4.1.1.6. Other Features

Still other features can be extracted for frames and regions of frames by computing their *texture*. Textural features are surprisingly useful in differentiating between video of natural and artificial objects (Picard and Liu 1994).

4.4.1.2. Computable Mid-Level Features

There have been significant advances in recent years in the automatic computation of mid-level features in video data. Using the low-level features described above, current signal-based parsing techniques have achieved reliable shot boundary detection (including cuts, fades, and dissolves), can detect pans and zooms in many cases, and within constrained video can track objects and recognize faces.

4.4.1.2.1. Video Segmentation

When we first began developing Media Streams, we used an average color value shot boundary detection algorithm. Today we use the algorithm developed by Nagasaka and Tanaka who evaluated and integrated many of the existing alternative approaches in developing their own (Nagasaka and Tanaka 1992). This algorithm calculates color histograms for 16 regions in each frame, throws out the 8 largest values, and then compares the 8 remaining color histograms for each frame by squaring their differences and normalizing (using the χ^2 test).

Current work in video segmentation is adding to the repertoire of tools by focusing on techniques which combine multiple analyses to recognize different types of shot boundaries. Researchers are developing algorithms for detecting fades and dissolves (Zhang and others 1993), scene boundaries (Hampapur and others 1994), news segment boundaries (Swanberg and others 1993), and television commercial boundaries (Williams 1994).

4.4.1.2.2. Camera Motion

Using optical flow techniques, researchers have made progress in recognizing pans and zooms (Akutsu and others 1992; Teodosio 1992; Tonomura and others 1993; Ueda and others 1991). The automatic recognition of more sophisticated camera motions is an open and active area of research (Tomasi and Kanade 1992).

4.4.1.2.3. Object Motion

The challenge of automatically tracking objects in motion sequences has achieved some success in constrained video (Ueda and others 1993) and researchers are working on object motion tracking in unstructured video as well (Woodfill 1992; Zabih and others 1993).

4.4.1.2.4. Object Recognition

Object recognition has had the most success in recognizing faces in constrained video (Pentland and others 1993a) and in differentiating objects according to their texture (Pentland and others 1994b).

4.4.2. Audio

Research is also being conducted in automatic segmentation and tagging of audio data by means of parsing the audio track for pauses and voice intensities (Arons 1993a), other audio cues including sounds made by the recording devices themselves (Pincever 1990), as well as specialized audio parsers for music, laughter, and other highly distinct acoustic phenomena (Hawley 1993). Advances in signal separation and speech recognition will also go a long way to automating the parsing of the content of the audio track. Media Streams currently uses a set of fixed thresholds for separating out speech from background noise from silence. Though this technique looks only at amplitude and uses fixed thresholds, it performs surprisingly well. In the future, we will incorporate the superior techniques described

below for detecting pauses in the audio track and separating out various types of acoustic events.

4.4.2.1. Pause Breaks

Significant work has been done by Barry Arons on pause detection and audio and speech parsing in general (Arons 1993a); we hope to incorporate these results into our system. Arons' work uses dynamic thresholding and windowing techniques to facilitate better detection of pauses in speech and the separation of speech from background noise in unstructured audio recordings.

4.4.2.2. Specialized Audio Parsers

Work done by Michael Hawley in developing specialized audio recognizers for musical events, speech, and click-lick sounds (e.g., footsteps) could be applied to automatically parsing the structure and enriching the representation of the audio track (Hawley 1993). There has also been some very recent work in separating voices of different speakers (Reynolds 1992; Reynolds 1994) and detecting the beat of music in the audio track (Goto and Muraoka 1994; Rosenthal 1992). This work looks promising for creating useful segmentations of audio data that Media Streams' content representation could make use of.

4.4.3. Why Automatic Representation is Not Enough

The achievements in automatic parsing of video and audio data outlined above form a very useful foundation for media annotation systems. The integration of such parsers into video annotation systems is clearly a necessary step in creating useful annotations not only for the labor that they save, but also for the types of phenomena they enable us to describe. Media Streams currently uses the Nagaska algorithm for shot boundary detection and fixed thresholds for audio pause detection. The incorporation of more advanced signal-based parsing techniques will enhance the system's ability to provide useful segmentations of video and audio data. However, though the parsers outlined above provide useful data, they do not generate a *sufficient* set of annotations for reusable media.

The results of automatic parsers do not capture semantic level information about video content. Specifically, they do not have the ability to differentiate the types of semantically invariant, sequence-independent features from semantically variable, sequence-dependent features. The

knowledge required to perform this categorization relies on many forms of common sense knowledge. Strangely, it takes a huge degree of contextual knowledge in order to be able to separate out those features of video that can be decontextualized. There is hope for automatic parsing of the motions of physical bodies in video, but the recognition that a scene contains no defining clues as to its being shot in Paris is a task requiring “AI complete” common sense knowledge that currently resides only in humans. The challenge is to design systems which leverage off the respective strengths of human and machine processing.

It is important to engage in a *Gedankenexperiment* about automatic representation in order to understand its theoretical limits and possibilities. Imagine if you will that in the next century we invent a machine that is capable of automatically parsing video and audio content. What would a complete parse mean? If we assume that this machine could attain human level performance in creating Media Streams annotations, it would have to have access to our common cultural knowledge. To use this common sense in recognizing and annotating video content, the machine would not only have to have this knowledge stored but have strategies for using this knowledge in the recognition and annotation task. These strategies would necessarily imply that the machine had the ability to interpret these cultural (and cinematic) common sensicals in concrete contexts of use. In order to do so, such a machine would have to have a *stance*, or a synthesis of various stances, in our cultural world. This notion of a stance is a phenomenological one. It is the site of our situated living activity of being-in-the-world. It is our stake in things which concern us and our vantage point from which the world takes on its meanings (Heidegger ; Merleau-Ponty 1962; Winograd and Flores 1986). A machine which had a stance would, in effect, be a participant in our common cultural experience.

If such a machine were to exist, the question of its relationship to human cultural production, especially the creation of representations of human cultural artifacts, would come to the fore. If this machine had a unique individual stance, it would be like a person. If it could somehow embody a multitude of stances, as a sort of living common cultural memory, then it would function as a community of humans. At that point the question of the need for particularly human representations would truly arise. If we could build a machine with human or trans-human intelligence, then we or it could build a machine with superhuman intelligence. Would a society with superhuman intelligences still feel the need for specifically human representations of video? If the particularity of human intelligence was superseded, as it might well be in a society that had a plethora of superhuman machine intelligences, the need for human annotation, and perhaps sadly the need for humans, might be gone entirely. Interestingly, Vernor Vinge, who thinks about these kinds of things, seems intrigued by

the possibility that superhuman intelligences might use *movies* to talk to each other—so *they* would be creating representations of video content (Vinge 1994).

Short of the advent of superhuman machine intelligence (and our subsequent demise), fully independent annotation of video by fully automatic processes will not be feasible. What is feasible and desirable is the description of media by humans and machines working as a functional assemblage, or as Donna Haraway describes, as a *cyborg*:

A cyborg is a cybernetic organism, a hybrid of machine and organism [...]. (Haraway 1991: 149).

We must completely reconceive the respective roles of machines and humans in the task of representing video. In film analysis as in video representation for retrieval and repurposing, we must combine the abilities of machines and humans into a representing cyborg that will make use of the power and speed of current and future machine parsing with the knowledge and cultural common sense of human beings. A human/machine cyborg leverages the respective skills of its functional members and compensates for their respective limitations wherever possible. *The cyborg notion reorders the functional parts and relations of human/machine representational activity.* This notion has ramifications not just for archives and film studies, but also for artificial intelligence, signal processing, and human computer interface design. We need to think about the design of “human in the loop” or “mixed initiative” algorithms in which the algorithms we write actually have interleaved human and machine computation as algorithmic steps. This strategy is not common within computer vision, but will result in the creation of far more powerful hybrid systems than strictly automated systems. The strategy is to leverage the algorithmic design off the respective strengths and weaknesses of humans and computers.

As an example, imagine we want to construct a parser whose task it is to find and track the characters in home videos. Traditional computer vision divides this task into segmentation, recognition, and tracking. To design this task as a human in the loop algorithm we find that the tasks which are currently very hard for a computer (recognition of which blobs are humans and of which humans they are), are simple for a human, and the tasks which are tedious and often inaccurately performed by a human (tracking blobs moving over time), are best performed by a computer. Recognition is a matter of seconds for a human, while tracking would be tedious; but for a computer, tracking is tractable and efficient. A mixed initiative algorithm would have the following steps:

- 1) computer finds blobs with faces that move
- 2) human classifies which blobs are people and which people they are
- 3) computer refines the contours of these blobs
- 4) human corrects any errors caused by occlusion, etc.
- 5) computer tracks these now segmented and classified people throughout the video sequence

The entire algorithm can be correctly thought of as having some computation done by silicon chips and some done by human cells. This hybrid functional unity of human and machine could create accurate annotations of the segments in which certain characters appear in a home video with greater speed and accuracy than either a human or machine working alone.

4.5. Representation for Retrieval and Repurposing

Whether done by human, machine, or cyborg, the central question we have addressed is the representation of video content for retrieval and repurposing. This representation is stream-based and uses a semantic ontology of composable terms for creating physically-based descriptions of those aspects of video content which are semantically invariant and sequence-independent. Through the process of retrieval and repurposing, Media Streams supports the representation of the semantically variable, sequence-dependent aspects of video content. The retrieval algorithms which use these representations to retrieve and repurpose video are described in the next chapter.



Chapter Five

Retrieving and Repurposing Video

5. Retrieving and Repurposing Video

5

Media Streams makes use of all the insights outlined above about knowledge representation for video. With an iconic visual language designed for video representation, users create stream-based representations of video content. Media Streams utilizes a hierarchically structured semantic space of iconic primitives which are combined to form compound descriptors which are then used to create multi-layered, temporally indexed annotations of video content. These iconic primitives are grouped into descriptive categories designed for video representation and are structured to deal with the special semantic and syntactic properties of video data. The categories include: space, time, weather, characters, objects, character actions, object actions, relative position, screen position, recording medium, cinematography, shot transitions, and subjective thoughts about the material.

Media Streams also takes into account the semantics and syntax of video in the design of its memory structures and retrieval algorithms. We have built a system that can retrieve video according to semantically invariant, sequence-independent descriptions of its content in order to repurpose that content through the construction of new sequences. Indexing in the memory structures aids in retrieval by adding a level of representation in the semantic hierarchy that captures the context-specific relations of terms. In Media Streams, the act of retrieval is itself an act of repurposing through retrieval-by-composition methods. In our retrieval system, we also enable the user to capture the semantically variable, sequence-dependent content of video through the creation of additional context specific indices which add new structure to the system's representations. By enabling the encoding of these two representations of content Media Streams offers a form of human-machine (cyborg) memory that overcomes the limitations of human-only representation and retrieval systems while preserving much of the subtlety and flexibility of human memory.

In this chapter we discuss memory, similarity, retrieval, and repurposing in Media Streams. Throughout our discussion we provide detailed examples of Media Streams' representational structures.

5.1. Memory and Retrieval

Current video retrieval systems, even if they use computational representations such as a database of keyword descriptors, rely for the most part on human memory for retrieval. Keywords may, if one is very lucky, direct an operator to a tape or reel which may contain possible hits, but it is the operator's memory of the footage and the associations which connect these memories which form the real mechanisms of representation and retrieval. However, human memory lacks the sharability and durability necessary for managing large scale video archives. Current artificial memory is durable and sharable but lacks structures which would encode semantically and analogically relevant relations to guide retrieval. Therefore, in attempting to create a video representation language which humans and machines can use, the challenge is to structure the representation of video in ways functionally similar to human memory without losing the performance of artificial memory. The goal is to create a combined system of human and artificial memory such that humans and machines can make use of each other's representations and machines can store and operate on associations similar to those that human beings use in memory and retrieval.

Such a cyborg system could create a memory that is trans-personal, not just a memory for an individual, or a small group of individuals, but a memory for an entire culture, the components of which are the video, audio, and of course textual artifacts which the culture produces. This has been a long-standing project since ancient times: the library of Alexandria, the memory palaces of the Renaissance and Baroque periods, the various digital libraries projects of today. The challenge for the age of digital media is to create a cyborg memory that enables us not only to organize and categorize, but to recombine and mutate our cultural artifacts into new forms. Media Streams is an effort to create an enabling technology that transforms the library into a movie studio, metamorphosing the archival resources of the past into the computational artifacts of the future.

5.1.1. Semantic and Episodic Memory

Within theories of human memory, an important distinction is made by Endel Tulving and others between *semantic* and *episodic* memory (Baddeley 1984; Tulving 1993) which provides a framework for thinking about how to structure the cyborg memory described above. Semantic memory can be thought of as the categorical or definitional part of human memory: remembering what a thing is and what class or category it belongs to. Tulving describes semantic memory in this way:

Semantic memory registers and stores knowledge about the world in the broadest sense and makes it available for retrieval. If a person knows something that is in principle describable in the propositional form, that something belongs to the domain of semantic memory. (Tulving 1993: 67).

Episodic memory can be thought of as the concrete recollection of a sequence of events, an episode. Tulving describes episodic memory as follows:

Episodic memory enables a person to remember personally experienced events as such. That is, it makes it possible for a person to be consciously aware of an earlier experience in a certain situation at a certain time. (Tulving 1993: 67).

To illustrate the difference between semantic and episodic memory, let us think of a chair. Semantic memory is what enables us to remember that a chair is a piece of furniture. Episodic memory is what enables us to remember a particular instance of interacting with a chair, like sitting in a particularly comfortable chair at my friend's house. The status of the functional and developmental relationships between semantic and episodic memory systems is a matter of some disagreement among various researchers. Tulving sees episodic memory as developing out of semantic memory:

The relation between episodic and semantic memory is hierarchical: Episodic memory has evolved out of, but many of its operations remain dependent on, semantic memory. A corollary is that semantic memory can operate (store and retrieve information) independently of episodic memory, but not vice versa. (Tulving 1993: 67-68).

Baddeley sees the reverse:

It also seems likely that items that are now in semantic memory were first represented as individual episodes. Consider for example the French word for *salt*; this was probably told to you in a French lesson at school. If you were questioned about it that evening, then attempting to recall it would almost certainly have relied on episodic memory. By now, if you remember it at all, it seems likely that it is a result of a wide range of subsequent episodes which you can probably no longer recall. (Baddeley 1984: 17).

Baddeley's position that semantic memory is built out of episodic memory through codification or categorization of repeated concrete experiences seems the more plausible one for human development because of the lack of a need for innate cognitive structures prior to any experience. However, implementing such a model with its requirement for generalizing and learning concepts from examples is an unsolved area in AI research. In our computational implementation, we begin with semantic memory structures and, as Tulving suggests, build our episodic memory structures on top of them. Semantic memory can function independently of episodic memory while episodic relies on the structure of semantic memory as a scaffolding. Semantic retrieval can function independently, as it does in our system when creating Icon Palettes through queries in the Icon Space. When retrieving video sequences in the Media Time Line, the system uses both semantic and episodic memory structures. It is when they work together that semantic and episodic memory structures provide a robust mechanism for the computational representation, storage, and retrieval of video data. Semantic memory structures provide a way of capturing the atemporal, categorical, sequence-independent semantics of video; episodic memory structures provide a way of representing the temporal, sequence-dependent, relational semantics of video. By combining both representations, we can develop a flexible yet robust semantics of descriptors and descriptions, and a storage and retrieval system that combines the strengths of human and artificial memory.

5.1.2. Media Streams Memory Structures

The underlying representation and semantics of the memory structures in Media Streams make use of three distinct organizational structures:

- *CIDIS*—the hierarchically structured semantic space of atemporal descriptors
- *Media Time Lines* — the relationally-structured syntactic space of temporal descriptions
- *Indices*—the indexing of relations on the CIDIS in the CIDI tree which were created on Media Time Lines

In order to understand the structure and function of these three forms of organization and how they enable us to create semantic and episodic memory structures we need to look more closely at their implementation in FRAMER and work through an example of their use.

5.1.2.1. Framer

FRAMER is the knowledge representation and database language in which Media Streams' structures for video representation are written. FRAMER was conceived of and developed by Prof. Kenneth Haase in order to provide a persistent framework for media annotation and description that supports cross-platform knowledge representation and database functionality (Haase 1994). Media Streams makes use of three of FRAMER's core functionalities:

- persistent object storage
- simple prototype-based inheritance
- indefinitely recursive annotation

FRAMER has other two other unique features—its own SCHEME-like extension language (FRAXL) and a non-deterministic interpreter incorporating McCarthy's AMB operator—which Media Streams does not make use of.

FRAMER has a few simple and elegant representational structures. The basic unit is called a *frame*. Frames have other frames beneath them called *annotations* (like directories in a UNIX file system). Frames can be indefinitely annotated by other frames which can be indefinitely annotated by other frames, and so on. The frame above an annotation is called its *home*. All frames are first class objects. Frames can have *grounds* which point to other frames or domain objects (e.g., filenames, lists, vectors, bitmaps, etc.). As an example, let's represent the knowledge that Fido the Wonder Dog has four legs and is a canine, which is a mammal, which is an animal. I can create a FRAMER structure with *animals* at the top and then have successive frame annotations to get me down to *Fido the Wonder Dog*. I can then annotate the *Fido the Wonder Dog* frame with the frame *legs* and give that frame a ground with a value of 4 in it:

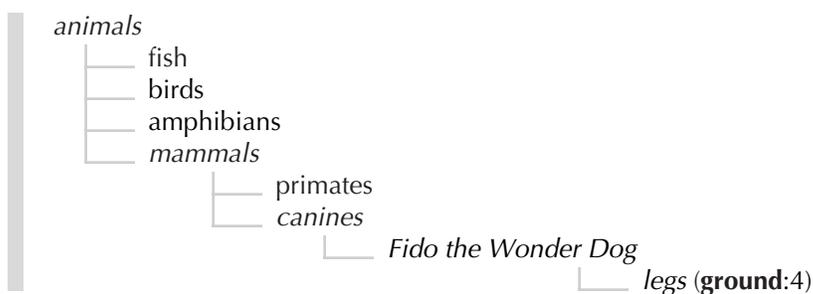


Figure 19. FRAMER Structure for Fido the Wonder Dog's legs

The home of the *Fido the Wonder Dog* frame is the *canines* frame whose home is the *mammals* frame whose annotations are the frames *primates* and *canines*. The *Fido the Wonder Dog* frame has as an annotation the frame *legs* whose ground is 4. FRAMER can save and restore large databases of frames and frames can be made, deleted, and edited on the fly.

FRAMER has a simple and pervasive way of expressing inheritance relations between frames: *prototypes* and *spinoffs*. If frame A is the *prototype* of frame B, then frame B is the *spinoff* of frame A. Prototypes and spinoffs obey the Default Prototype Rule:

Default Prototype Rule:

GIVEN
A is an annotation of X
A' is an annotation of Y

IF
X is a prototype of Y

THEN
A is a prototype of A'

For example, if *Fido the Wonder Dog* is the prototype of *Spot* and has the annotation *legs*, and *Spot* also has the annotation *legs*, then by the Default Prototype Rule, the annotation *legs* whose home is *Spot* has the annotation *legs* whose home is *Fido the Wonder Dog* as its prototype. Prototype relations enable frames to inherit information from one another. If we extend the world of Fido the Wonder Dog to include our new dog, Rover, by creating a Rover frame under the frame Pets and by making Rover's prototype Fido the Wonder Dog, I can then ask Rover if he has legs by calling the function probe-annotation:

(probe-annotation #/Pets/Rover "legs")

Which would return the now inherited frame for Rover's legs:

#/Pets/Rover/legs

If I ask for the ground of Rover's legs, it will be empty, but if I ask for the inherited ground of Rover's legs, I will find out he has 4, just like Fido the Wonder Dog, his prototype.

FRAMER's use of prototype relations is informed by important work in knowledge representation and cognitive science that might loosely be called "prototype theory." The idea in prototype theory, whose foundations George Lakoff articulates in contradistinction to objectivist epistemologies (Lakoff 1987) is that humans (and one hopes one day machines) organize and develop their understanding of the world not by fitting perceptions to a set of abstract categories, but by means of analogy to prototypical examples. The perception of a particular chair is not organized as an instance of the concept of "chair" that is a subconcept of the class of "furniture" but rather is related to a particular prototypical (often early) perception of another particular chair.

The development of semantic memory structures out of episodic memory structures could be understood as the codification of prototypical examples from concrete episodes. These emergent prototypes would serve as the nodal points of a network of relationships between prototypes and their spinoffs. New experiences would be understood in terms of the background of old experiences as represented by the network of prototypes. New experiences would then be indexed in the appropriate places in the network of prototypes. Media Streams' own indexing structures and mechanisms work much in this way since they are built on top of *Mnemosyne*, an analogy matcher and indexing and retrieval program written on top of FRAMER by Prof. Kenneth Haase (Haase 1991; Haase 1993).

Prototype theory has also informed pioneering work in programming language design. Early work in object-oriented languages had a more flexible, prototype-based organization than the now conventional class-instance relationship (Winograd 1978). In a prototype-based system, the class-instance distinction does not exist—any object can form a "class" by being the prototypical object of a group of objects which are related to it even in non-uniform ways. This type of prototype-based organization also has a certain analogy to the family-resemblance structure of categorization described in the late Wittgenstein (Wittgenstein 1958).

With annotations, prototypes, and grounds, FRAMER enables us to create intricate structures and paths of inheritance among frames. There are then three distinct types of relational structures in FRAMER which can be used independently or be interwoven in a representational design:

- **annotation hierarchy**
(frames and their annotations)
- **prototype network**
(frames and their spinoffs)
- **ground pointers**
(frames and their grounds)

Media Streams makes use of all of these structures in representing video content. The annotation hierarchy is used throughout since it is the fundamental structure in FRAMER, but it is used specifically as a representation mechanism in the deep hierarchies of the CIDIS. The CIDIS also form a network of prototype relations among themselves that often corresponds to but occasionally diverges from the annotation hierarchy in order to achieve something like multiple inheritance and to structure indexing in the CIDIS. Prototype relations also connect annotations on Media Time Lines to their prototypes in the CIDIS. CIDIS are the prototypical atemporal semantic descriptors out of which all descriptions are made within the system. On Media Time Lines, descriptions are made out of CIDI spinoffs, their relations, and their temporal extents. Grounds enable annotations to point to each other on Media Time Lines so as to express case frame like relations. Grounds and prototypes are the main structures used in indexing the occurrences of annotations on Media Time Lines within the structure of the CIDIS.

In order to clarify Media Streams' use of FRAMER structures and to illustrate the relationships between CIDIS, Media Time Lines, and Indices which enable Media Streams to combine the functions of semantic and episodic memory, we will work through an example of annotating, indexing, and retrieving a segment of video.

5.1.2.2. Example: Maya Lying on a Beach

Here is a shot of Maya Deren, an adult female. Maya is lying on a beach while waves crash into her. The location is the threshold (or border) of a beach; the objects present are waves and the beach on which Maya lays.



Figure 20. A shot of Maya Deren lying on a beach

5.1.2.2.1. CIDIS

In order to annotate this shot of Maya we need to get the appropriate iconic descriptors from the Icon Space. The Icon Space is the interface to the CIDIS, the hierarchically structured semantic space of atemporal descriptors, which we need to access. Let's first get an iconic descriptor for the character Maya Deren.

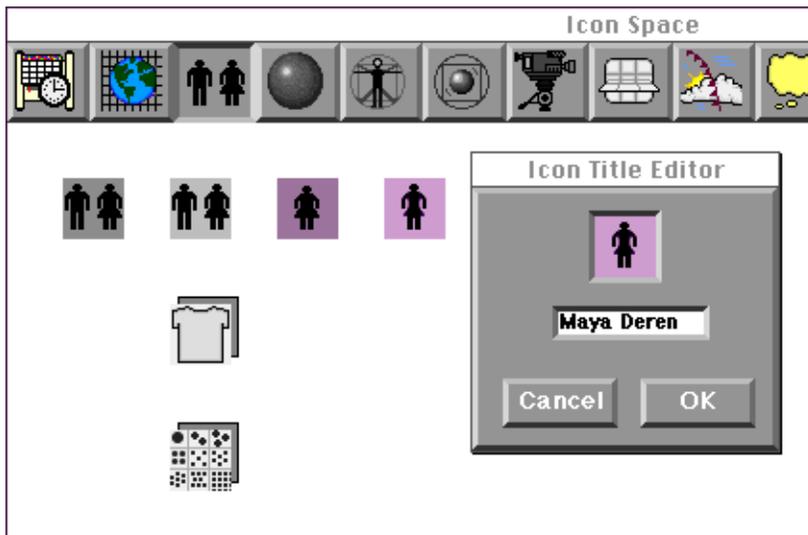


Figure 21. Constructing an icon for Maya Deren in the Icon Workshop

In the CIDIS, this iconic descriptor for Maya Deren is represented by a *frame* with its *annotations* and *prototypes*. We can look at the FRAMER structure to see how Maya Deren is represented. Below is the output of DI (a browser for FRAMER frames) for the frame that is Maya Deren's *home* and *prototype*. Let's walk through it together.

The frame described below is *adult female*.

```
#/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female
;Description of "adult female" <17>
```

Adult-female's home is *female* whose home is *character* whose home is *characters* whose home is *CIDIS* whose home is *MEDIA-STREAMS* whose home is the root frame *#/*.

```
; >> in "female" <60> in "character" <248> in "characters" <249> in "CIDIS" <250> in "MEDIA-STREAMS" <2> in #/ <3>
```

Adult-female's prototype is *female* whose prototype is *character of indeterminate sex*, which is a terminal prototype. A terminal prototype is a frame that has spinoffs and is not the spinoff of any other frame.

```
; >> like #/MEDIA-STREAMS/CIDIS/characters/character/female <60>
; like #/MEDIA-STREAMS/CIDIS/characters/character/character\ of\ indeterminate\ sex <61>
```

Adult-female has no ground.

```
; >> with no grounding
```

Adult-female has 26 annotations. All of which except for the last two have prototypes, expressed in DI by *...like<frame>*.

```
; >> with 26 annotations:
```

These annotations express the types of default relations that spinoffs of *adult-female* can have to other frames on Media Time Lines. These annotations function much like slots in a more traditional knowledge representation language.

```
; > action <73>          .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/action <252>
; > numbers <262>       .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/numbers <263>
; > occupation <272>    .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/occupation <273>
; > screen position <278> .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/screen\ position <279>
; > object <264>        .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/object <265>
; > relative position <282> .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/relative\ position <283>
; > subject <284>      .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/subject <285>
```

The *SELF* annotation is always present. The identity relation it expresses is used in retrieval when a spinoff of a CIDI on a Media Time Line has no relations to any other frames.

```
; > SELF <89>          .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/SELF <267>
```

These annotations are frames that were created when a user made a titled version of *adult-female*, thus extending the CIDI hierarchy. *Maya Deren* is frame 255 and has *adult-female* as her prototype and as her home.

```
; > Melinda <254>      .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female <17>
; > Maya Deren <255>   .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female <17>
; > marilyn <274>     .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female <17>
; > Janet Cahn <280>  .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female <17>
; > Juliana Hatfield <281> .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female <17>
```

This *+CLOS-OBJECT* annotation is called a “+annotation” in FRAMER and is ignored in certain operations. +annotations are useful for adding information that is not inherited or inheritable. +annotations are a type of annotation; annotations without a “+” in front of them are also referred to as “features”. This +annotation stores in its ground a pointer to the CLOS object that *adult-female* is represented by in the Media Streams user interface.

```
; > +CLOS-OBJECT <75> .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/+CLOS-OBJECT <258>
                          ground= #<adult female cidi> <259>
```

This *+COMPOUNDS* annotation contains a list in its ground of the unique compound icons which use *adult-female* as a component.

```
; > +COMPOUNDS <275>  ... like #/MEDIA-STREAMS/CIDIS/characters/character/female/+COMPOUNDS <276>
                          ground=  ( #/MEDIA-STREAMS/COMPOUND-ICON-INDEX/adult\ female-dive
                                     #/MEDIA-STREAMS/COMPOUND-ICON-INDEX/adult\ female-lie.1
                                     #/MEDIA-STREAMS/COMPOUND-ICON-INDEX/adult\ female.2) <277>
```

This *+SUBORDINATE-ICONS* annotation contains a list in its ground of the subordinate icons which should appear below *adult-female* in the Icon Workshop. This ground is empty. If we were to put *Maya Deren* in the ground of the *+SUBORDINATE-ICONS* annotation, she would appear as an icon in the Icon Workshop below *adult-female*.

```
; > +SUBORDINATE-ICONS <260> .... (no prototype)
                          ground= nil <261>
```

This *+ICON-NAME* annotation contains the name of the icon used to display *adult-female* in the Media Streams user interface:

```
; > +ICON-NAME <286> .... (no prototype)
                                ground= "adult female" <287>
```

Among the many things that the FRAMER structure above expresses, it represents the following atemporal semantic knowledge about Maya Deren (what appears in parentheses is implied but not explicitly stated in the representation):

Annotation Hierarchy:

Maya Deren is an (instance of) *adult female*
adult female is a (subclass of) *female*
female is a (subclass of) *character*

Prototype Network:

Maya Deren is an (instance of an) *adult female*
adult female is a (type of a) *female*
female is a (specialization of a) *character of indeterminate sex*

The other icons which we need to annotate the shot of Maya Deren we obtain from the Icon Workshop:

The **character action**:



Figure 22. A path down the character-action hierarchy to *lying*

The action is lying (which is an action of the body being in contact with a surface). The FRAMER frame for the CIDI is:

```
#/MEDIA-STREAMS/CIDIS/characters\ actions\one-person\ action/body\ action/stand/lie
```

The **objects**:

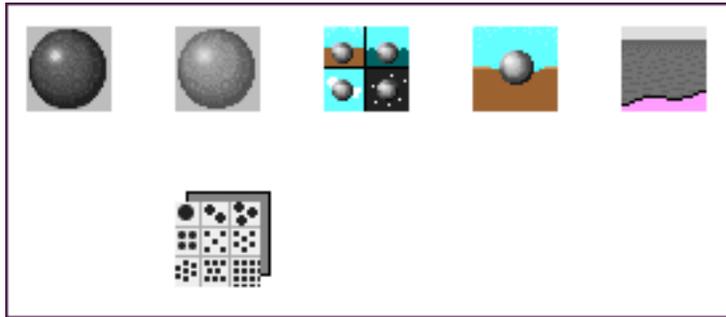


Figure 23. A path down the objects hierarchy to *beach*

A beach is a natural land object. The FRAMER frame for the CIDI is:

```
#/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ land\ object/beach
```

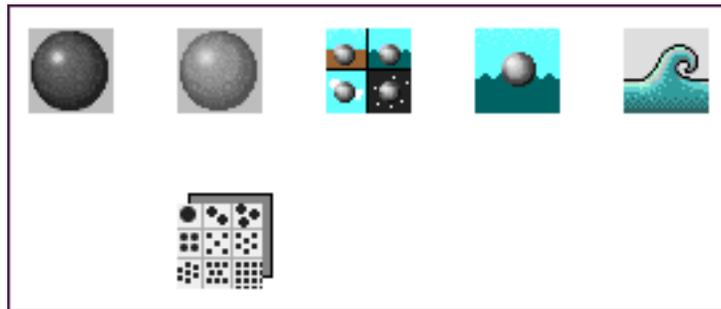


Figure 24. A path down the objects hierarchy to *wave*

A wave is a natural aquatic object. The FRAMER frame for the CIDI is:

```
#/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ aquatic\ object/wave
```

The **object action**:



Figure 25. A path down the object-action hierarchy to *wave-crash*

A wave crash is a liquid object action. The FRAMER frame for the CIDI is:

```
#/MEDIA-STREAMS/CIDIS/objects\ actions/single-object\ action/single-object\ state\ transformation/action\ involving\ liquid/liquid\ object\ action/wave\ crash
```

The **spatial location**:

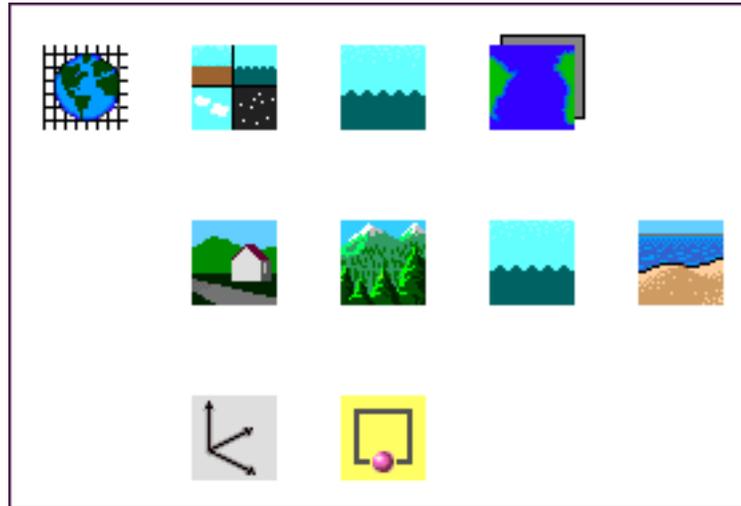


Figure 26. Paths down the spatial-location hierarchy to *ocean, beach* and *on-the-threshold-of*

The scene is set on the threshold of a beach on an ocean. The FRAMER frames for the CIDIS which form the spatial location compound are:

```
#/MEDIA-STREAMS/CIDIS/space/geographic\ location/sea/ocean  
#/MEDIA-STREAMS/CIDIS/space/functional\ space/wilderness/water\ wilderness/beach  
#/MEDIA-STREAMS/CIDIS/space/topographical\ location/on\ the\ threshold\ of
```

At this stage in annotation we have accessed those CIDIS we need to annotate the shot. The CIDIS have a structure closely resembling that of Tulving's descriptions of semantic memory: categorical, definitional, atemporal, semantic. For the CIDIS, the annotation of the shot of Maya Deren can be thought of as a list of descriptors from a semantic hierarchy without semantic or temporal relations: Maya Deren, lying, beach, waves, waves crash, on the threshold of a beach on an ocean.

Let us now turn to using these CIDIS to annotate the shot of Maya Deren on a Media Time Line, the relationally structured syntactic space of temporal descriptions.

5.1.2.2.2. Media Time Lines

In the Media Time Line, spinoffs of the CIDIS are given temporal extents (in and out points) and semantic relations which connect them into an episodic structure. In the Media Time Line below we have annotated the annotation streams for *visually inferable spatial location*, *character*, *character action*, *object*, and *object action* with the icons we gathered from the Icon Space. The glommed icon (“Maya lying on a beach”) used in the *character action* annotation stream and the glommed icon (“A wave crashing into Maya”) used in the *object action* annotation stream were created by successively dropping their component icons onto the Media Time Line.

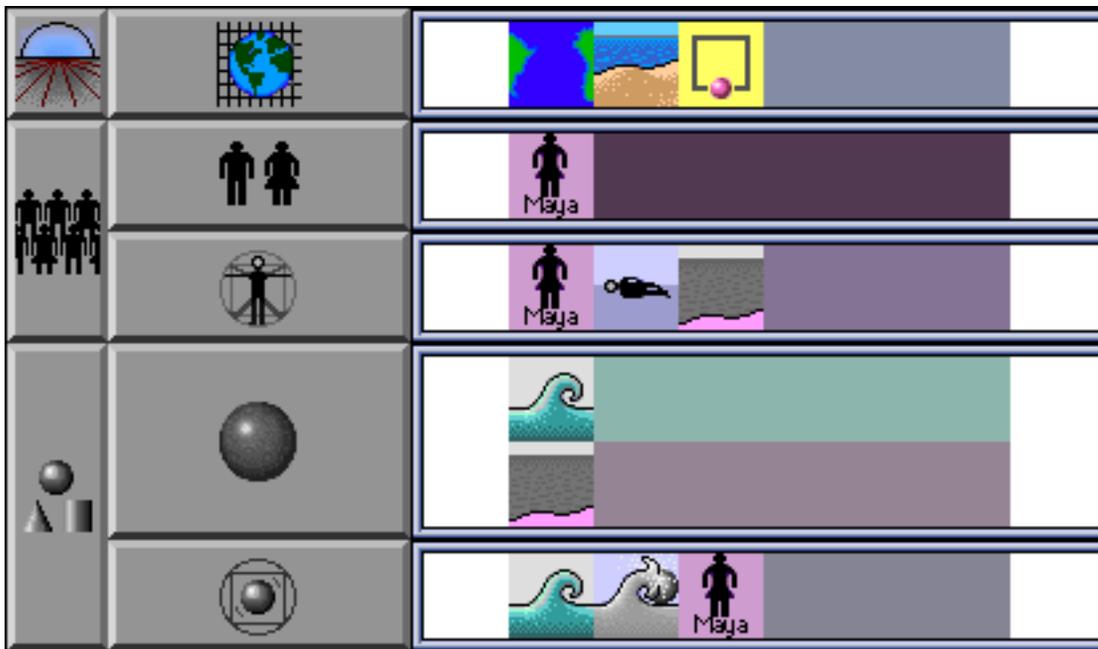


Figure 27. A Media Time Line describing the shot from Figure 20

The FRAMER structure for the Media Time Line shows the semantic and temporal relations created between the CIDIS spinoffs through the process of annotation. Let’s take a closer look.

The Media Time Line frame described below is named by its unique creation date: *12-01-1994@14:19:15*.

```
#/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15
;Description of "12-01-1994@14:19:15" <0>
```

12-01-1994@14:19:15's home is *MEDIA-TIME-LINES* whose home is *MEDIA-STREAMS* whose home is the root frame #/.

```
; >> in "MEDIA-TIME-LINES" <1> in "MEDIA-STREAMS" <2> in #/ <3>
```

12-01-1994@14:19:15 has no ground and no prototype.

```
; >> with no grounding
```

12-01-1994@14:19:15 has 15 annotations (12 features and 3 +annotations).

```
; >> with 15 annotations:
```

These 12 annotations are spinoffs of the CIDIS which are their prototypes.

```
; > wave crash <146>      .... like #/MEDIA-STREAMS/CIDIS/objects\ actions/single-object\ action/single-object\ state\
                           transformation/action\ involving\ liquid/liquid\ object\ action/wave\ crash <147>
; > Maya Deren.2 <148>    .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/Maya\ Deren <9>
; > wave.1 <149>         .... like #/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ aquatic\ object/wave <150>
; > Maya Deren <151>     .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/Maya\ Deren <9>
; > wave <152>           .... like #/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ aquatic\ object/wave <150>
; > lie <4>              .... like #/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/stand/lie <5>
; > beach.1 <6>         .... like #/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ land\ object/beach <7>
; > Maya Deren.1 <8>    .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/Maya\ Deren <9>
; > beach <10>          .... like #/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ land\ object/beach <7>
; > on the threshold of <299>.... like #/MEDIA-STREAMS/CIDIS/space/topographical\ location/on\ the\ threshold\ of <300>
; > beach.2 <301>      .... like #/MEDIA-STREAMS/CIDIS/space/functional\ space/wilderness/water\ wilderness/beach <302>
; > ocean <303>        .... like #/MEDIA-STREAMS/CIDIS/space/geographic\ location/sea/ocean <304>
```

This *+CLOS-OBJECT* annotation stores in its ground a pointer to the CLOS object that *12-01-1994@14:19:15* is represented by in the Media Streams user interface.

```
; > +CLOS-OBJECT <12>    .... (no prototype)
                           ground= #<media-time-line #x59CBB79> <13>
```

This *+NAME* annotation contains the name of the window used to display *12-01-1994@14:19:1* in the Media Streams user interface.

```
; > +NAME <14>      .... (no prototype)
                        ground= "Maya on the Beach" <15>
```

This *+SCENES* annotation has no prototype and no ground. If we had been annotating a video sequence with multiple shots, this FRAMER structure would have a series of SCENE annotations, each of which stores in its ground a list of the Media Time Line annotations (CIDI spinoffs) which were valid for the temporal extent of that SCENE.

```
; > +SCENES <16>      .... (no prototype)
```

Let's look now at how the Media Time Line FRAMER structures express the idea that "Maya is lying on a beach." We first can look at the FRAMER structure for *Maya Deren.1* which functions as the subject of the action of "lying on a beach" and is a spinoff of *Maya Deren* in the CIDIS. The frame described below is *Maya Deren.1*.

```
#/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/Maya\ Deren.1
;Description of "Maya Deren.1" <8>
```

Maya Deren.1's home is *12-01-1994@14:19:15* whose home is *MEDIA-TIME-LINES* whose home is *MEDIA-STREAMS* whose home is the root frame #/.

```
; >> in "12-01-1994@14:19:15" <0> in "MEDIA-TIME-LINES" <1> in "MEDIA-STREAMS" <2>in #/ <3>
```

Maya Deren.1's prototype is *Maya Deren* whose prototype is *adult female* whose prototype is *female* whose prototype is *character of indeterminate sex*, which is a terminal prototype meaning that it is not the spinoff of any other prototype.

```
; >> like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/Maya\ Deren <9>
; like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female <59>
; like #/MEDIA-STREAMS/CIDIS/characters/character/female <60>
; like #/MEDIA-STREAMS/CIDIS/characters/character/character\ of\ indeterminate\ sex <61>
```

Maya Deren.1 has no ground.

```
; >> with no grounding
```

Maya Deren.1 has 12 annotations (3 features and 9 +annotations).

; >> with 12 annotations:

The *SELF* annotation is always present. The identity relation it expresses is used in retrieval when a spinoff of a CIDI on a Media Time Line has no relations to any other frames.

; > SELF <74> like #MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female\ Maya\ Deren\ SELF <75>
ground= #MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15\ Maya\ Deren.1 <8>

This *action* annotation relates *Maya Deren.1* to the frame *lie* of which *Maya Deren.1* is the subject by storing a pointer to the frame *lie* in the ground of *action*. The prototype of *action* is the *action* annotation of *Maya Deren* in the CIDIS.

; > action <64> like #MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female\ Maya\ Deren\ action <65>
ground= #MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15\ lie <4>

This *object* annotation relates *Maya Deren.1* to the frame *beach.1* which is the object of *Maya Deren.1*'s lying action. A pointer to the frame *beach.1* in the ground of *object*. The prototype of *object* is the *object* annotation of *Maya Deren* in the CIDIS.

; > object <62> like #MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female\ Maya\ Deren\ object <63>
ground= #MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15\ beach.1 <6>

The 9 +annotations store information about the following things: the close-object that represents *Maya Deren.1* in the Media Time Line (in this case the glommed icon for "Maya Deren is lying on a beach"); the creation-date when the frame *Maya Deren.1* was created; the time-line-context (annotation stream) in which *Maya Deren.1* occurs; the row in the annotation stream in which the annotation for *Maya Deren.1* occurs; the color of the color bar for the *Maya Deren.1* annotation; the end-frame at which the extent of the *Maya Deren.1* annotation ends; the start-frame at which the extent of the *Maya Deren.1* annotation begins; the logger who made the *Maya Deren.1* annotation; and the entry in the compound-icon-index which corresponds to the unique compound icon in which *Maya Deren.1* appears.

```

; > +CLOS-OBJECT <66>.... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/Maya\ Deren/+CLOS-
      OBJECT <67>
      ground= #<characters-action-compound-icon #x5A0D301> <20>
; > +CREATION-DATE <68> .... (no prototype)
      ground= "12-01-1994@14:19:45" <26>
; > +TIME-LINE-CONTEXT <69>.... (no prototype)
      ground= "characters actions.v" <28>
; > +ROW <70>      .... (no prototype)
      ground= 0 <30>
; > +COLOR <71>    .... (no prototype)
      ground= 8549777 <32>
; > +END-FRAME <72>.... (no prototype)
      ground= 66 <163>
; > +START-FRAME <73>.... (no prototype)
      ground= -13 <165>
; > +LOGGER <76>   .... (no prototype)
      ground= #/MEDIA-STREAMS/COMPOUND-ICON-INDEX/adult\ male-annotator.1 <39>
; > +COMPOUND-ICON-INDEX <77> .... (no prototype)
      ground= #/MEDIA-STREAMS/COMPOUND-ICON-INDEX/adult\ female-lie-beach <41>

```

Like the FRAMER structure for *Maya Deren.1*, the FRAMER structures below for *lie* and *beach.1* express the subject/action/object relationships between the annotations *Maya Deren.1*, *lie*, and *beach.1*.

```

#/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/lie
;Description of "lie" <4>
; >> in "12-01-1994@14:19:15" <0> in "MEDIA-TIME-LINES" <1> in "MEDIA-STREAMS" <2>in #/ <3>
; >> like #/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/stand/lie <5>
; like #/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/stand <17>
; >> with no grounding
; >> with 12 annotations:
; > SELF <36>      .... like #/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/stand/lie/SELF <37>
      ground= #/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/lie <4>
; > object <21>    .... like #/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/stand/lie/object <22>
      ground= #/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/beach.1 <6>
; > subject <23>   .... like #/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/stand/lie/subject <24>
      ground= #/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/Maya\ Deren.1 <8>
; > +CREATION-DATE <25> .... (no prototype)
      ground= "12-01-1994@14:19:45" <26>
; > +TIME-LINE-CONTEXT <27> .... (no prototype)
      ground= "characters actions.v" <28>
; > +ROW <29>      .... (no prototype)
      ground= 0 <30>
; > +COLOR <31>    .... (no prototype)
      ground= 8549777 <32>
; > +END-FRAME <33> .... (no prototype)
      ground= 66 <163>
; > +START-FRAME <35>.... (no prototype)
      ground= -13 <165>

```

```

; > +LOGGER <38>      .... (no prototype)
                        ground= #/MEDIA-STREAMS/COMPOUND-ICON-INDEX/adult\ male-annotator.1 <39>
; > +CLOS-OBJECT <18> .... like #/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\
                        action/stand/lie/+CLOS-OBJECT <19>
                        ground= #<characters-action-compound-icon #x5A0D301> <20>
; > +COMPOUND-ICON-INDEX <40> .... (no prototype)
                        ground= #/MEDIA-STREAMS/COMPOUND-ICON-INDEX/adult\ female-lie-beach <41>

#/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/beach.1
; Description of "beach.1" <6>
; >> in "12-01-1994@14:19:15" <0> in "MEDIA-TIME-LINES" <1> in "MEDIA-STREAMS" <2> in #/ <3>
; >> like #/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ land\ object/beach <7>
; like #/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ land\ object <42>
; >> with no grounding
; >> with 12 annotations:
; > SELF <55>          .... like #/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ land\ object/beach/SELF <56>
                        ground= #/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/beach.1 <6>
; > action <43>        .... like #/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ land\ object/beach/action <44>
                        ground= #/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/lie <4>
; > subject <45>       .... like #/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ land\ object/beach/subject <46>
                        ground= #/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/Maya\ Deren.1 <8>
; > +CLOS-OBJECT <47> .... like #/MEDIA-STREAMS/CIDIS/objects/object/natural\ object/natural\ land\ object/beach/+CLOS-
                        OBJECT <48>
                        ground= #<characters-action-compound-icon #x5A0D301> <20>
; > +CREATION-DATE <49> .... (no prototype)
                        ground= "12-01-1994@14:19:45" <26>
; > +TIME-LINE-CONTEXT <50> .... (no prototype)
                        ground= "characters actions.v" <28>
; > +ROW <51>          .... (no prototype)
                        ground= 0 <30>
; > +COLOR <52>        .... (no prototype)
                        ground= 8549777 <32>
; > +END-FRAME <53>    .... (no prototype)
                        ground= 66 <163>
; > +START-FRAME <54> .... (no prototype)
                        ground= -13 <165>
; > +LOGGER <57>      .... (no prototype)
                        ground= #/MEDIA-STREAMS/COMPOUND-ICON-INDEX/adult\ male-annotator.1 <39>
; > +COMPOUND-ICON-INDEX <58> .... (no prototype)
                        ground= #/MEDIA-STREAMS/COMPOUND-ICON-INDEX/adult\ female-lie-beach <41>

```

The **prototype network** of the Media Time Line expresses which frames on the Media Time Line are spinoffs of prototypes in the CIDIS. The **ground pointers** enable the Media Time Line to express various *relations* (subject, action, object, etc.) between the frames.

At this stage in annotation, we have expressed various semantic and temporal relations between the CIDIS spinoffs we used to annotate the shot. The annotations of the Media Time Line have a structure closely resembling that of Tulving's descriptions of episodic memory: relational, temporal, syntactic. The temporal relations in a Media Time Line are expressed implicitly in the various *+start-frame* and *+end-frame* annotations. We used to store and index the symbolic representations of temporal relations for every Media Time Line annotation, but it turned out to be considerably more efficient to compute temporal relations at query time than to store and retrieve them. For the Media Time Line, the annotations of the shot of Maya Deren can be thought of as a paragraph of sentences, a structured description which adds semantic, syntactic, and temporal relations to descriptors from a semantic hierarchy: Maya Deren (an adult female), a beach (a natural land object), and waves (a natural aquatic object) are in the shot; Maya Deren is the subject of an action of lying whose object is a beach; the waves are the subject of an action of waves crashing whose object is Maya Deren; the location is the threshold of a beach on an ocean.

Let us now turn to seeing how the relations of the Media Time Line are indexed in the CIDIS. Indexing extends the semantic memory of the CIDIS through the encoding of episodic knowledge and provides the framework for retrieval and repurposing.

5.1.2.2.3. Indices

Indexing is the process of creating structures which link the Media Time Lines' sequence-dependent, episodic relations between the spinoffs of CIDIS to their prototypes in the semantic hierarchy of the CIDIS. In effect, indexing connects episodic and semantic memory. Let's take a closer look to see how this is done in our FRAMER structures by examining the indices which represent that Maya Deren (and all her prototypes) can be the subject of the action of lying (and all its prototypes).

The frame described below is *action*, which is the action annotation on the Maya Deren frame in the CIDIS which is the prototype of the Maya Deren.1 frame which appeared in the Media Time Line. The *action* frame is itself the prototype of the action annotation on the Maya Deren.1 frame in the Media Time Line (remember the Default Prototype Rule applies here: if Maya Deren has a spinoff Maya Deren.1, and Maya Deren has an annotation *action*, then that action annotation will be the prototype of the action annotation of Maya Deren.1).

```

#/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/Maya\ Deren/action
;   Description of "action" <65>
;   >> in "Maya Deren" <9> in "adult female" <59> in "female" <60> in "character" <80> in "characters" <81> in "CIDIS" <82> in
    "MEDIA-STREAMS" <2>in #/ <3>
;   >> like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action <95>
;   like #/MEDIA-STREAMS/CIDIS/characters/character/female/action <96>
;   like #/MEDIA-STREAMS/CIDIS/characters/character/action <97>
;   >> with no grounding

```

The *action* frame has a *+index* annotation which stores in its ground a list which binds the various action indices and the terminal prototypes of the actions they index. The ground of the *+index* annotation functions as a kind of index of the indices which are stored under *+index*.

```

;   >> with 1 annotations:
;   > +index <98>      .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index <99>
                        ground= {(#/MEDIA-STREAMS/CIDIS/objects\ actions/single-object\ action/single-object\ state\
                        transformation/action\ involving\ liquid . #^/+index/v.13)
(#/MEDIA-STREAMS/CIDIS/objects\ actions/single-object\ action/single-object\ motion/move\ in\ a\ straight\ line . #^/+index/v.12)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/head\ action/eye\ action . #^/+index/v.11)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/arm\ and/or\ hand\ action/hand\ action . #^/+index/v.10)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/head\ action/head\ action/abstract\ head\ action . #^/+index/v.9)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/abstract\ body\ action/turn\ about\ the\ waist .
    #^/+index/v.8)

```

Here is the cons pair which binds the terminal prototype *stand* of the *lie* CIDI (*/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/stand*) with the index (*/^/+index/v.7*) which stores all the action annotations of Maya Deren spinoffs on Media Time Lines whose grounds have the *stand* CIDI as their terminal prototype. Indexing groups relations on Media Time Lines under their prototypes in the CIDIS according to the common terminal prototypes of their grounds.

```

(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/stand . #^/+index/v.7)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/abstract\ body\ action/turn\ counter\ clockwise .
    #^/+index/v.6)
(#/MEDIA-STREAMS/CIDIS/objects\ actions/action\ involving\ two\ objects/two-object\ motion/move\ through . #^/+index/v.5)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/wear . #^/+index/v.4)
(#/MEDIA-STREAMS/CIDIS/cinematography/tripod\ cinematography . #^/+index/v.3)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/sit . #^/+index/v.1)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/walk . #^/+index/v.2)} <100>

```

If we look at the annotations of the *+index* frame, we see the action annotations of Maya Derens from Media Time Lines stored under their respective indices.

```

#/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/Maya\ Deren/action/+index
;   Description of "+index" <98>
;   >> in "action" <65> in "Maya Deren" <9> in "adult female" <59> in "female" <60> in "character" <80> in "characters" <81> in
;   "CIDIS" <82> in "MEDIA-STREAMS" <2> in #/ <3>
;   >> like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index <99>
;   like #/MEDIA-STREAMS/CIDIS/characters/character/female/action/+index <101>
;   >> with grounding { see above } <100>
;   >> with 15 annotations:
;   > v.13 <102>      .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.13 <103>
;                       ground= #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.1/action <104>
;   > v.12 <105>      ... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.12 <106>
;                       ground= #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.24/action <107>
;   > v.11 <108>      .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.11 <109>
;                       ground= {#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.27/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.26/action} <110>
;   > v.10 <111>      .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.10 <112>
;                       ground= {#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.20/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.23/action} <113>
;   > v.9 <114>       .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.9 <115>
;                       ground= #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.73/action <116>
;   > v.8 <117>       .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.8 <118>
;                       ground= #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.74/action <119>

```

Here is the index (#^/+index/v.7) which stores all the action annotations on Maya Deren spinoffs on Media Time Lines whose grounds share the terminal prototype *stand* in the CIDIS. In this index we find the action annotation from our Media Time Line *12-01-1994@14:19:15* of Maya lying on the beach whose ground is *lie*: #/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/Maya\ Deren.1/action.

```

;   > v.7 <120>      .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.7 <121>
;                       ground= {#/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/Maya\ Deren.1/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.2/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.11/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.17/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.8/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.14/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.5/action} <122>
;   > v.6 <123>      .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.6 <124>
;                       ground= #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.75/action <125>
;   > v.5 <126>      .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.5 <127>
;                       ground= {#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.40/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.35/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.46/action} <128>
;   > v.4 <129>      .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.4 <130>
;                       ground= {#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.48/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.34/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.18/action
;                       #/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.9/action

```

```

# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.15/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.12/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.42/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.36/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.6/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.3/action} <131>
; > v.3 <132>
.... like # /MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.3 <133>
ground= {# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.76/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.79/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.78/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.77/action} <134>
; > v.2 <135>
.... like # /MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.2 <136>
ground= {# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.49/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.33/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.30/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.43/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.50/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.29/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.38/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.32/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.31/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.44/action
# /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.37/action} <137>
; > v.1 <138>
.... like # /MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v.1 <139>
ground= # /MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\ Deren.19/action <140>
; > +v.count <141>
.... like # /MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/+v.count <142>
ground= 13 <143>
; > v <144>
.... like # /MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index/v <145>

```

The prototypes of Maya Deren in the CIDIS contain all of the CIDI Maya Deren's indices plus their own indices and the indices of their spinoffs. In this *+index* frame we find indexed all of the action annotations on Media Time Lines whose prototypes are the action annotation of *adult-female* in the CIDIS.

```

# /MEDIA-STREAMS/CIDIS/characters/character/female/adult\ female/action/+index
; Description of "+index" <99>
; >> in "action" <95> in "adult female" <59> in "female" <60> in "character" <80> in "characters" <81> in "CIDIS" <82> in "MEDIA-STREAMS" <2> in #/ <3>
; >> like # /MEDIA-STREAMS/CIDIS/characters/character/female/action/+index <101>
; >> with grounding {(# /MEDIA-STREAMS/CIDIS/objects\ actions/single-object\ action/single-object\ state\ transformation/action\ involving\ liquid. ^/v.17)
(# /MEDIA-STREAMS/CIDIS/objects\ actions/single-object\ action/single-object\ motion/move\ in\ a\ straight\ line . #^/index/v.16)
(# /MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/head\ action/eye\ action. #^/index/v.15)
(# /MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/head\ action/head\ action/abstract\ head\ action . #^/index/v.14)
(# /MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/abstract\ body\ action/turn\ about\ the\ waist . #^/index/v.13)

```

Here is the cons pair which binds the terminal prototype *stand* (`#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/stand`) with the index (`#{^/+index/v.12}`) which stores all the action annotations of Maya Deren spinoffs on Media Time Lines whose grounds have the *stand* CIDI as their terminal prototype.

```
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/stand . #{^/index/v.12)
(#/MEDIA-STREAMS/CIDIS/objects\ actions/action\ involving\ two\ objects/two-object\ motion/move\ through . #{^/index/v.11)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/walk . #{^/index/v.10)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/sit . #{^/index/v.9)
(#/MEDIA-STREAMS/CIDIS/objects\ actions/action\ involving\ two\ objects/two-object\ state\ transformations/reflected\ in .
  #{^/index/v.8)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/wear . #{^/index/v.7)
(#/MEDIA-STREAMS/CIDIS/cinematography/tripod\ cinematography . #{^/index/v.6)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/body\ action/abstract\ body\ action/turn\ counter\ clockwise
  #{^/index/v.5)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/head\ action/mouth\ action/make\ sound\ with\ mouth .
  #{^/index/v.4)
(#/MEDIA-STREAMS/CIDIS/characters\ actions/one-person\ action/arm\ and/or\ hand\ action/hand\ action . #{^/index/v.3)
(#/MEDIA-STREAMS/CIDIS/objects\ actions/single-object\ action/single-object\ state\ transformation/sheet\ action .
  #{^/index/v.1)
  (#/MEDIA-STREAMS/CIDIS/cinematography/lens\ cinematography . #{^/index/v.2}) <304>
;   >> with 19 annotations:
```

We have elided all other indices to show the index (`#{^/+index/v.12}`) which stores all the action annotations on spinoffs of *adult-female* on Media Time Lines whose grounds share the terminal prototype *stand* in the CIDIS. In this index we still find the action annotation from our Media Time Line *12-01-1994@14:19:15* of Maya lying on the beach whose ground is *lie*: `#/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/Maya\Deren.1/action`. We also find an action annotation for another spinoff of *adult-female*, *marilyn.1*, from Media Time Line *09-07-1994@14:48:18*: `#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@14:48:18/marilyn.1/action`.

```
;   > v.12 <106>      .... like #/MEDIA-STREAMS/CIDIS/characters/character/female/action/+index/v.12 <319>
ground= {#/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/Maya\Deren.1/action
#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@14:48:18/marilyn.1/action
#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\Deren.2/action
#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\Deren.11/action
#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\Deren.17/action
#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\Deren.8/action
#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\Deren.14/action
#/MEDIA-STREAMS/MEDIA-TIME-LINES/09-07-1994@13:09:27/Maya\Deren.5/action} <320>
```

If we were to look at the index under the subject annotation of *lie* in the CIDIS, we would find an index of subject annotations of spinoffs of *lie* from Media Time Lines whose grounds have the common terminal prototype: character of indeterminate sex. Within this index of subject annotations, we would find the annotation whose ground is: `#/MEDIA-STREAMS/MEDIA-TIME-LINES/12-01-1994@14:19:15/Maya\ Deren.1`.

Indices reciprocally index relations from Media Time Lines under their respective prototype terminals in the CIDIS. Prototype terminals, the top nodes of prototype chains in the CIDIS, serve to create groupings in the semantic memory. By indexing annotations from Media Time Lines in these groupings, we group the parts of episodic examples into their “natural kinds”. Prototype terminals articulate the boundaries of semantic categories in the representational structures of Media Streams. When the system needs to find contextual examples from Media Time Lines of the various terms expressed in the semantic memory, indexing has placed them in the appropriate places for retrieval.

For the Indices, the annotations of the shot of Maya Deren can be thought of as memories of the episode of “Maya lying on the beach” located in the appropriate parts of the semantic memory. I remember... “Maya lying on a beach” translates to the indices of: Maya can be the subject of a lying action; an adult female can be the subject of a lying action; a character of indeterminate sex can be the subject of a lying action; and a lying action can have Maya as its subject; a beach can be the object of a lying action, etc.

Now that we understand how the content of video is represented semantically, episodically, and indexed in Media Streams’ memory structures, let us turn to examining what types of retrieval we might want to perform on video and the retrieval mechanisms in Media Streams which make use of the CIDIS, Media Time Lines, and Indices.

5.2. Similarity and Retrieval

Semantic and episodic memory structures enable us to create a mixed representational system that can answer the fundamental problem of video retrieval systems: how do we determine the similarity of descriptors, of descriptions, of shots, and of sequences? Similarity needs to be context-sensitive and compare not just descriptors, but relations between them. The determination of similarity holds the key to retrieval, and due to the properties of video as a medium (especially its semantic and syntactic features discussed above) the semantic and episodic memory systems must work together in order to retrieve video based on its unique features.

We can investigate some of the various types of similarity a video retrieval system needs to accommodate by looking at the example of trying to retrieve a video segment of “a hammer hitting a nail into a piece of wood”. Imagine I have the following candidate matches for my query:

- video of a hammer hitting a nail into a piece of wood
- video of a hammer, a nail, and a piece of wood
- video of a nail hitting a hammer, and a piece of wood
- video of a sledgehammer hitting a spike into a railroad tie
- video of a rock hitting a nail into a piece of wood
- video of a hammer swinging
- video of a nail in a piece of wood

How do we rank these candidates in terms of their similarity to the query? Clearly the “video of a hammer hitting a nail into a piece of wood” is the best match since it is an exact match. That is easy. But what about the other candidates? We can differentiate three forms of similarity which are useful here: *semantic*, *relational*, and *temporal*. Semantic similarity is the similarity of terms in a semantic hierarchy normally expressed as the distance along a valid path between terms. Relational similarity is the similarity of the syntactic structure of the relations between these terms. Temporal similarity is the similarity of temporal relationships among relations and terms. In order to understand these types of similarity, let’s reexamine the above candidates and see how they satisfy these three types of similarity.

The “video of a hammer, a nail, and a piece of wood” has exact semantic and temporal similarity, but has no relational similarity.

The “video of a nail hitting a hammer, and a piece of wood” has exact semantic and temporal similarity, but has incorrect relational similarity.

The “video of a sledgehammer hitting a spike into a railroad tie” has approximate semantic similarity of the subject and objects of the action and exact semantic similarity of the action. It has exact temporal and relational similarity.

The last three candidates are more complex and more interesting.

The “video of a rock hitting a nail into a piece of wood” has exact relational and temporal similarity, but has incorrect semantic similarity because, at least in Media Streams’ semantic hierarchy, a rock can only reach a hammer by a long and crooked path. One could imagine, though, that through indexing, a particular example of a rock being the subject of a nail hitting action could be found by a much more direct path under the index of a hammer being the subject of nail hitting action. What this index would express is that though in most cases a rock is not a hammer or even semantically similar to one in the ontology, in this particular case a particular rock has *functioned as* a hammer in a nail hitting action so is *functionally similar* to a hammer. Media Streams provides exactly this kind of extension mechanism for indexing through the ability to *shift* and *reindex* the prototypes of specific grounds of relations from Media Time Lines to functionally similar prototypes in the CIDIS.

The “video of a hammer swinging” and the “video of a nail in a piece of wood” respectively have exact but incomplete semantic, relational, and temporal similarity. But what if in asking for a “video of a hammer hitting a nail into a piece of wood” a system could retrieve these two video segments in *sequence*? A sequence of these two segments would have exact and complete semantic similarity, exact but incomplete relational similarity, and a series of temporal relations between the terms which if they were treated as all existing in one shot, would be dissimilar to those in the query. In the query, the hammer, nail, hitting, and piece of wood all are contemporaneous. In this synthetic result sequence, the temporal relations include simultaneity of some but not all of the terms and sequentiality among the rest. We see in this sequence the creation of a satisfactory result through the montage of less satisfactory partial matches. A central challenge for video retrieval systems is to be able to construct this sequence of partial matches as a result and to be able to score it in relation to the other candidates such that the satisfactoriness of this solution is greater than the sum of its parts.

Media Streams employs all three types of similarity to compare video annotations:

- **Semantic Similarity**
Similarity as expressed by various types of distance between CIDIS
- **Relational Similarity**
Similarity as expressed by the relations of annotations on Media Time Lines and the indexing of the grounds of those relations on their prototypes in the CIDIS
- **Temporal Similarity**
Similarity as expressed by the temporal relations calculated between annotations on Media Time Lines. The temporal extents of annotations can satisfy these temporal relations exactly or be coerced to satisfy them either through subsampling of the temporal extents of the annotations (temporal clipping) or through the relaxation of the temporal constraints these temporal relations imply (temporal relaxation).

In Media Streams, the similarity of a given description to another can be determined based on their and their components' semantic similarity, the similarity of their relations, their structures, and their grounds, and the temporal similarity of their calculated temporal relations. The retrieval algorithms we have developed to use these three forms of similarity are discussed within the next section.

If we think about the candidates we discussed in the example of retrieving matches to a video segment of "a hammer hitting a nail into a piece of wood" we will find that Media Streams offers new forms of retrieval which enable us to use our semantically invariant, sequence-independent representations of video content to retrieve sequences which have a variable, sequence-dependent semantics. Furthermore, in the synthesis of a new sequence out of our last two candidates we see the kernel of a notion of video sequence retrieval that can use our representations not only to *find* existing sequences, but to *make* them in response to queries. The video representation language itself, the three forms of memory structures (CIDIS, Media Time Lines, and Indices), and the use of algorithms which can compare annotations according to the three types of similarity (semantic, relational, and temporal) make possible a new type of video retrieval that transforms retrieval into an act of montage and repurposing.

5.3. Repurposing and Retrieval

Video databases require new types of content representation to support retrieval. Traditional notions of the retrieval process itself need to be rethought for databases of audio and video. The process of retrieval in Media Streams, and I would argue in many of the applications for which video databases will be used in the future, intimately combines *finding* with *making*. The purpose of doing a query is not merely to find an existing piece of information as in a textual or numerical query, but to find or assemble matches for use in the construction of new sequences. To paraphrase Ben Dubrovsky, retrieval is traditionally conceived as “the termination of a search process rather than as the beginning of a composition process” (Dubrovsky 1991). In Media Streams, video is not conceived of as a collection of atomic nodes or objects but as streams of data rich descriptions which can yield many possible objects, even construct them by combining existing segments of video into new sequences.

These sequences would not be limited to existing sequences in the archive but could be assembled from shots taken from different video sources based on their respective and combined similarity to the given query. This way the combinatorics of video resources expands to not only the shots and sequences indexed in the archive, but to all of their possible combinations and permutations.

Our research seeks to reorient the development of video retrieval technologies toward the facilitation of the repurposing of video content. In Media Streams, retrieval is an act of repurposing because of our retrieval-by-composition methods. Retrieval involves an act of montage. We seek to reinterpret the act of movie making along the lines of Eisenstein’s techne-centered understanding of composition:

“One does not create a work,” writes Eisenstein in his diary in 1919; “one constructs it with finished parts, like a machine. *Montage* is a beautiful word: it describes the process of constructing with prepared fragments.” (Bordwell 1993: 121).

By creating tools which use retrieval-by-composition methods we work towards solving the twofold needs of Garage Cinema makers: tools for accessing content and tools for manipulating content. Media Streams points toward a solution to the second challenge in terms of a solution for the first: the development of composition tools that enable users to repurpose video content through retrieval-by-composition methods, through interleaved acts of finding and making.

5.3.1. Media Streams Retrieval Mechanisms

We will now look more closely at the algorithms Media Streams uses to find similar video segments and to construct new sequences out of parts of existing ones.

5.3.1.1. Mnemosyne

The indexing and basic matching algorithms in Media Streams use *Mnemosyne*, an analogical knowledge representation system built on top of FRAMER by Professor Kenneth Haase (Chakravarthy and others 1992; Haase 1991; Haase 1993). “Mnemosyne” (named after the Greek goddess of memory who was also the mother of the nine muses) is a radically memory-based representational system in which analogical matching forms the core representation. Mnemosyne offers a very different paradigm of knowledge representation and the role of indexing and analogy in the formation of a dynamic memory. The challenge that this memory-based representation addresses is the inflexibility and brittleness of most semantic or categorical representations. In knowledge representations where a fixed semantic structure is not sufficient to allow flexibility of the representation, analogical memory-based representations are needed so that the meanings of the descriptors used in descriptions are, in effect, defined by their differences and similarities to concrete examples of their use.

In contrast to systems like CYC (Lenat and Guha 1990), in which a large body of canonical representation must take place *before* analogy matching can occur, in Mnemosyne, the “canonicalness” of a representation is expressed by the accretion and record of past analogizing between concrete examples indexed on prototypes (the trace memory). In contrast to Schank’s work on dynamic memory (Schank 1982) which gave rise to case-based reasoning techniques (Riesbeck and Schank 1989), Mnemosyne does not have a set of prior abstractions under which examples are indexed, but rather a base ontology of prototypes whose indices of their relations and use in concrete contexts form the basis of matching and generalization. Haase writes:

The knowledge has migrated into two places: the indexing structure and the matching mechanism. In Mnemosyne in particular, this means the prototypical relations between descriptions and sub-descriptions and the variations (and records of past analogizing) stored in the trace network. (Haase 1991: 5).

Mnemosyne’s ability to index and match examples of *relations* between descriptors under their common prototypes enables it to form an *episodic*

memory that indexes and thereby supports the comparison of descriptions according to their *relational similarity*. Media Streams makes use of this functionality in order to represent the semantically variable, sequence-dependent relations of representations of video content. By indexing the similarity of relations between CIDIS, as expressed on Media Time Lines, under the prototypes of these relations in the CIDIS, Media Streams uses Mnemosyne's mechanisms to build up analogical representations of the semantics of descriptors from contextual examples of their relation and use.

Mnemosyne is built on top of the three basic FRAMER structures described above. It uses the *annotation hierarchy* to build up descriptive components, the *prototype network* to indicate analogous components, and the *ground pointers* to describe relations between components. In Mnemosyne, FRAMER's prototype-spinoff relationship is used to create a *cognate* relation on which all matching is based. Haase defines the cognate relation as follows:

Two descriptive elements are cognates if there exists some prototype common to them which is common to no other pairings of other elements from the descriptions. (Haase 1991: 10).

In Media Streams, the grounds of relations are cognates. This allows our retrieval algorithm to avoid spending time trying to match components which should not be compared, as, for example, trying to match the subject of an action to the object of the same action.

Let us now look at Media Streams' retrieval algorithm that uses the indexing and matching functionality of Mnemosyne.

5.3.1.2. Media Streams Retrieval Algorithm

Media Streams' retrieval algorithm **get-similar-sequences** has two main steps for finding/making similar sequences:

- **find-similar-compound-icon-frames**
Iterate over the compounds in the query to generate a similarity array structure, which binds each compound in the query to a list of matching compounds, sorted in order of their semantic similarity to the compound in the query. This uses Mnemosyne indexing to limit the number of comparisons which need to be made.
- **find-best-templates**

Consider each explicitly marked shot of the query as a template, and find the best combination of matching compounds which satisfy the temporal relations within the shot to fill the template. Treat the problem as a graph search problem and find the path through the space of combinations with the highest total score. Return the best n matches for each shot. These matches can themselves be sequences of video segments taken from various Media Time Lines. Return the top n scoring combinations of shots.

Appendix B: Code Listing for Media Streams Retrieval Algorithm contains the major functions of the algorithm with extensive comments by Brian Williams. In this section we will discuss particular aspects of the two main steps of the algorithm in order to do two things: highlight the ways Media Streams calculates the similarity of Media Time Line annotations in terms of the relations which connect them and the structure of their prototype network and Indices (thus using semantic and relational similarity); and discuss the issues we faced and the solutions we came up with for concatenating matches drawn from multiple Media Time Lines into temporally consistent, matching video sequences (thus satisfying temporal similarity).

5.3.1.2.1. Computing Semantic and Relational Similarity

Within **find-similar-compound-icon-frames**, Media Streams' retrieval algorithm compares the *relations* between the components of compounds on Media Time Lines. By matching on relations (and their grounds which point to the components they relate), we enable our algorithm to use both relational and semantic similarity. The relations capture the sequence-dependent structure of Media Time Lines. The indices of these relations group accreted similar examples of these structures, while the prototypes capture the sequence-independent semantic similarity of the terms under which these relations are indexed.

When given a Media Time Line to use as a query, **find-similar-compound-icon-frames** takes all of the visible annotations within the desired temporal extent and iterates over them to find those features (non +annotations) with grounds. In Media Time Lines, the only features with grounds are *relations* like action, subject, object, etc., whose grounds are the components of the compounds the relations connect.

Once we find the relations (and their grounds) in the query Media Time Line, we use Mnemosyne-style matching to search through the prototypes

of these relations in the CIDIS in order to find relations with similar grounds pointed to by indexed occurrences of these relations on other Media Time Lines which are stored on the Indices of their common prototypes within the CIDIS. For example, if we want to find an occurrence on another Media Time Line of a “Maya lying on a beach” we would begin by searching in the Indices for similar grounds of the relations in our query Media Time Line. We could begin by looking for a match for the ground of the *subject* relation of the lie annotation (which is Maya Deren.1) in the Indices of the prototype of the subject relation in the CIDIS. The prototype of the subject relation of lie on the Media Time Line is the subject annotation of lie in the CIDIS.

We then look on its Indices to find occurrences of subjects performing lying actions on other Media Time Lines. Since the Indices are structured in such a way as to *index similar occurrences under their common terminal prototype*, all the indexed occurrences on Media Time Lines of subjects of lying actions which have “character of indeterminate sex” as their terminal prototype (Maya Deren.1 among them) will be stored in the same place in the Indices of the subject annotation of lie in the CIDIS.

After we have found the correct place in the Indices to begin our search (which is itself a trivial operation because of the prototype relations and the structure of the Indices themselves), we perform two operations:

- we score the similarity of the grounds of relations with a scoring function based on various types of distance in their prototype network
- we search up the prototypes of the relations in the CIDIS in order to find more matches in their Indices

The scoring function calculates a multi-part score of the similarity of any two CIDIS. The first element is the point value for being an **exact** match (the higher the better); the second element is number of steps in the CIDIS the *prototype* of the **query** is away from its spinoffs in the match (the lower the better); the third element is number of steps in the CIDIS the *prototype* of the **match** is away from its spinoffs in the query (the lower the better); the fourth element is the point value for being a **sibling** match, i.e., the match and the query are both the immediate spinoffs of a common prototype (the higher the better); and the fifth element is the point value for being a so-called **bad** match in which the query and match are not siblings but share a common-prototype (the higher the better). Scores are compared by ordered step-wise comparison of their partial scores.

The search up the prototype chains of the relations in the CIDIS terminates at their common terminal prototypes. If a query uses a spinoff of a CIDI that occurs above the common terminal prototypes (as in the case when queries use very general descriptors like “object” or “person”), our algorithm follows the annotation hierarchy down from such a descriptor in the CIDIS until a terminal prototype is reached (collecting any indexed matches on the way), and then begins the search out along the prototype network from that point. This design allows our algorithm not to have to compare everything to everything else (which it would if there was only a small number of terminal prototypes connecting all CIDIS), and to be able to start from very general terms and recur down to find similar matches in the prototype network.

As all matches are found and scored in the Indices, they are placed in a hash table binding each compound in the query to a list of matching compounds, sorted in order of their semantic similarity to the compound in the query. Once the matching process is complete, the next task is to construct sequences of the highest scoring matches which satisfy the temporal relations expressed in the query Media Time Line.

5.3.1.2.2. Computing Temporal Similarity

Within **find-best-templates**, Media Streams’ retrieval algorithm compares the *temporal relations* between the retrieved matching compounds in order to construct valid matching sequences. A template is the set of compounds in the query connected by their temporal relations. The temporal relations used in Media Streams to compare annotations are the 13 temporal relations articulated by James Allen (Allen 1985):

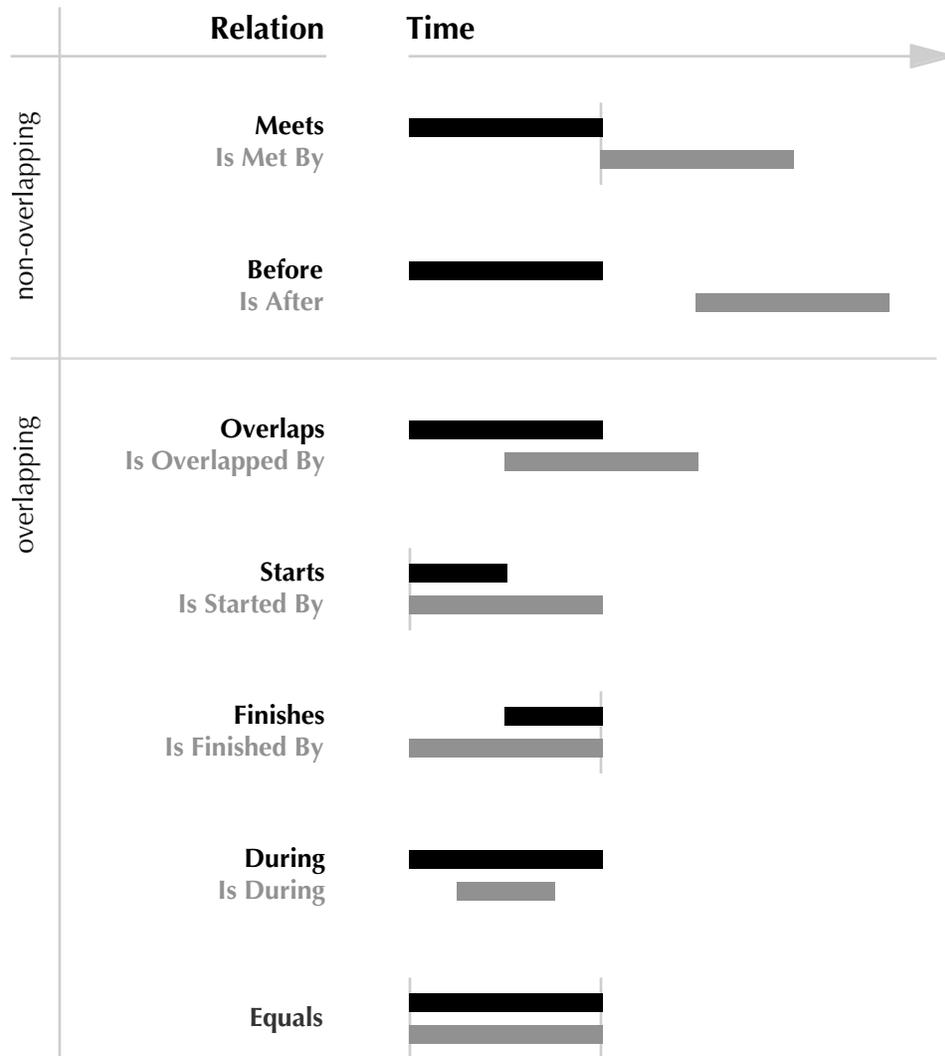


Figure 28. The 13 temporal relations

We group these relations into *temporally overlapping* (overlaps/is overlapped by, starts/is started by, finishes/is finished by, during/is during, and equals) and *temporally non-overlapping* relations (meets/is met by, before/after). This distinction is important for video, since temporally non-overlapping relations can often be usefully overridden through reordering of video shots.

The computation of temporal consistency is the heart of this step of the retrieval algorithm. The function **constraint-satisfied-by-path** performs the consistency checking of temporal constraints. It compares the temporal relationship between two compounds of the query to the temporal relationship between their matching compounds.

Media Streams moves beyond simple temporal consistency in several important ways. Temporal consistency is upheld if either of the compounds in the temporal relationship in the match is missing. This allows partial matches to still be temporally consistent. If a temporal relation can be calculated there are four forms of possible temporal consistency. If the temporal relation is the same as the temporal relation in the query, they are of course temporally consistent. If the temporal relation is a non-overlapping temporal relation we consider it satisfied since in our retrieval algorithm any non-overlapping temporal relation will match to any other non-overlapping temporal relation. This allows Media Streams to resegment and resequence parts of Media Time Lines in order to create satisfactory templates (new sequences). The other two cases extend the ability of the system to match overlapping temporal relations: temporal clipping and temporal relaxation.

5.3.1.2.2.1. Temporal Clipping

Temporal clipping coerces a temporal relation in the match by resetting the start and/or end frames of the grounds of the temporal relation. If the desired temporal relation is some overlapping relation, and the calculated temporal relation is an overlapping relation that can be clipped to become the desired temporal relation, clipping occurs. All prior temporal relations in the match are then rechecked for temporal consistency. If there is an inconsistency, the changes to the temporal extents of the match are undone.

5.3.1.2.2.2. Temporal Relaxation

Temporal relaxation allows certain less specific temporal relations to satisfy more specific temporal constraints. In Media Streams, if the desired temporal relation is *starts/is started by* or *finishes/is finished by*, it can be satisfied by the temporal relation *equals* in the match.

Using each retrieved compound as a node in a graph search problem, the algorithm finds the highest scoring temporally consistent paths through the graph. A completeness-threshold specifies how partial a path can be and still be considered a valid path. Once the number of completed, temporally consistent, best scoring templates reaches the matches-threshold (by default 10), these templates are assembled into video sequences and displayed to the user in sorted order.

We will next turn to an example in order to examine the types of retrieval functionality Media Streams' algorithm affords, namely the ability to retrieve video sequences by composing video segments which are

annotated by physically-based, semantically invariant, sequence-independent descriptors which are indexed so as to represent their semantically variable, sequence-dependent relations.

5.3.1.3. Retrieval-By-Composition Example: How Did She Get There?

If we return to our shot of Maya lying on the beach, we can imagine querying for a video sequence that would show us how she got there.



Figure 29. A shot of Maya Deren lying on the threshold of the beach as waves crash into her

Here is our original Media Time Line of the shot of Maya lying on the beach:



Figure 30. A Media Time Line describing the shot from Figures 29 and 20

Let's make a query for a sequence in which Maya dives off a cliff into the sea and then we see her lying on the beach. As we see in the Media Time Line below, the same interface we used for annotation is used for query, since annotation is describing video we have, while query is describing video we want to find/make. The sequence described below would add new meaning to our shot of Maya lying on the beach by using the shot either as the end of a "cliff diving" or a "jump to her death" sequence depending on the other video segments retrieved.

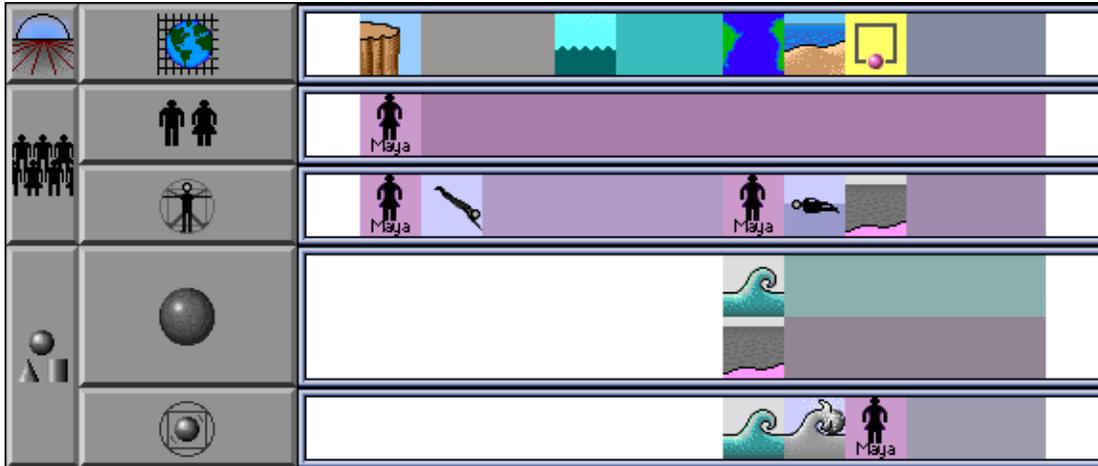


Figure 31. A Media Time Line describing footage we wish to find

Below is the Result Palette (the Icon Palette in the Icon Space displays the results of Media Time Line Queries) with the first 10 result sequences. In Media Streams, result sequences are represented by Media Time Line Icons similar to the VideoStreamer objects (developed by Eddie Elliott) which represent a video segment (and its length) by stacking video frames into an XY-T volume (Elliott 1993). Media Time Line Icons have two important innovations: they represent the extent of annotation (and thereby some notion of the density and dynamics of the content) of the video segment by representing the annotations of its Media Time Line in reduced form on the side face of the object; they also solve the problem of representing movies of widely differing lengths in the same window—Media Time Line Icons use *logarithmic* (as opposed to linear) scaling of the depth of the image volume to represent the length of the video segment.

The Result Palette shows the video sequences which were created to match the query in sorted order starting at the upper left corner and snaking down to the lower right corner. In this Result Palette, the first eight sequences are two shot sequences; the last two sequences have three shots each.



Figure 32. The result window produced by the query shown in Figure 31

The best matching sequence is represented below.

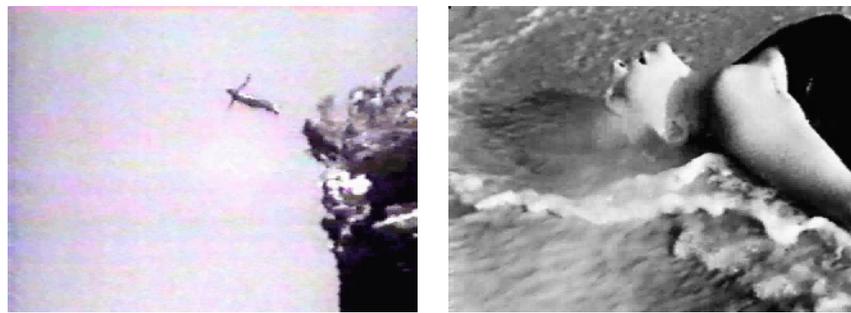


Figure 33. Stills taken from the best-matching result

The second shot in this sequence is the shot of “Maya lying on a beach”. The first shot (from a Rock Hudson movie) is of a character of indeterminate sex diving off a cliff into the sea. The database contained no shots of Maya diving off a cliff into the sea, but this wide shot in which the sex (and identity) of the character is indeterminate works perfectly well to create the “Maya diving off a cliff into the sea and washing up on a beach” sequence.

The reasons this sequence matched are explained by the *Score Window* below. The Score Window shows the compound icons of the query in the first column, the matching elements of the match in the second column, and the source Media Time Line, duration, and score of the match in the third column. The score is the multi-step score of the Media Streams Retrieval algorithm. The first element is the point value for being an **exact** match (the higher the better); the second element is number of steps in the CIDIS the *prototype* of the **query** is away from its spinoffs in the match (the lower the better); the third element is number of steps in the CIDIS the *prototype* of the **match** is away from its spinoffs in the query (the lower the better); the fourth element is the point value for being a **sibling** match, i.e., the match and the query are both the immediate spinoffs of a common

prototype (the higher the better); and the fifth element is the point value for being a so-called **bad** match in which the query and match are not siblings but share a common-prototype (the higher the better).

#<template #x6737481>		
QUERY	MATCH	
		at-land 00:14:09 - 00:26:03 #<score e q m s b t: 1 0 0 0 0>
		at-land 00:14:09 - 00:22:09 #<score e q m s b t: 5 0 0 0 0>
		at-land 00:00:00 - 00:27:15 #<score e q m s b t: 1 0 0 0 0>
		at-land 00:14:09 - 00:23:20 #<score e q m s b t: 5 0 0 0 0>
		at-land 00:13:05 - 00:59:24 #<score e q m s b t: 3 0 0 0 0>
		rock-hudson 00:24:25 - 00:28:08 #<score e q m s b t: 1 0 0 0 0>
		rock-hudson 00:24:25 - 00:29:01 #<score e q m s b t: 2 0 2 0 0>
		rock-hudson 00:28:09 - 00:32:21 #<score e q m s b t: 1 0 0 0 0>

Figure 34. The Score Window for the best-matching result

The next seven best matching sequences all have the same score. We will show three of them which offer interesting insights into the workings of retrieval and repurposing in Media Streams. The third sequence from the Result Palette is represented below.



Figure 35. Stills from the third sequence returned

The first shot is of Maya crawling on a table. The continuity of actor enabled the system to retrieve this first shot even though its action is similar to diving only by a bad match (since both are forms of human locomotion) and there is no continuity of location. Maya's route to the beach is a peculiar one in this sequence, but we do recognize that it is Maya who is making the journey. The reasons this sequence matched are shown in the Score Window below.

#<template #x6798481>		
QUERY	MATCH	
		at-land 00:14:09 - 00:26:03 #<score e q m s b t: 1 0 0 0 0>
		at-land 00:14:09 - 00:22:09 #<score e q m s b t: 5 0 0 0 0>
		at-land 00:00:00 - 00:27:15 #<score e q m s b t: 1 0 0 0 0>
		at-land 00:14:09 - 00:23:20 #<score e q m s b t: 5 0 0 0 0>
		at-land 00:13:05 - 00:59:24 #<score e q m s b t: 3 0 0 0 0>
		at-land 03:11:16 - 03:17:04 #<score e q m s b t: 2 0 0 0 1 0>

Figure 36. The Score Window for the third sequence returned

The fourth and fifth sequences from the Result Palette offer a different explanation of how Maya got to be lying on the beach.

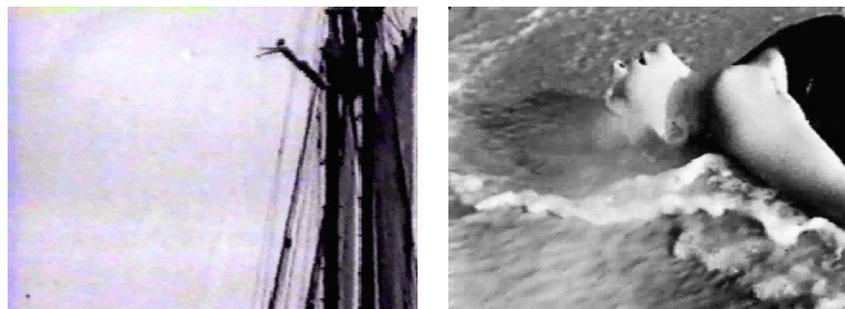


Figure 37. Stills from the fourth sequence returned

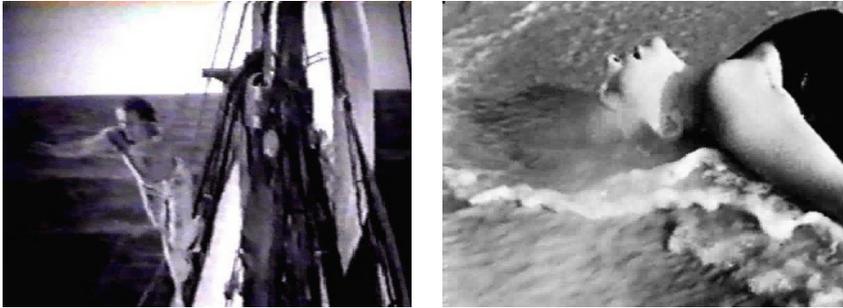


Figure 38. Stills from the fifth sequence returned

The first shots of these sequences match for the inverse of the reason that the first shot of the third sequence matched: they have an exact match on the action of diving but a bad match on the subject of the action. The fourth sequence appears to function better as a matching sequence since the wide framing of its first shots makes it less easy to determine the sex of the diver than in the medium framing of the first shot of the fifth sequence. If the query had specified that this portion of the sequence should be a wide shot, the fourth sequence would have scored higher than the fifth. As the query is currently formulated, both sequences received the same score. The reasons these sequences matched are shown in the Score Windows below.

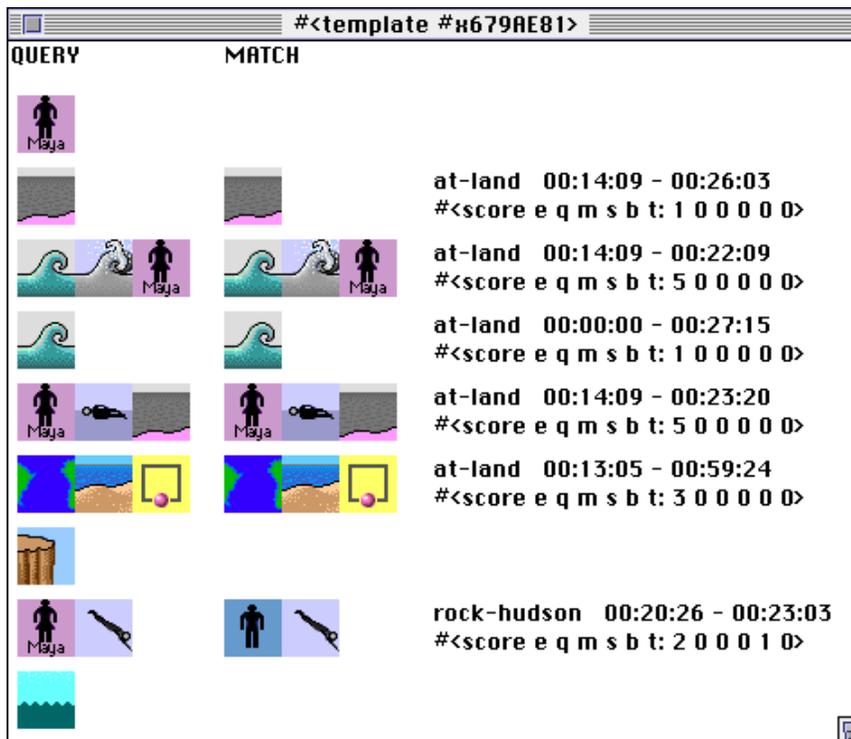


Figure 39. Score Window for the fourth sequence returned

#<template #H679B2B9>	
QUERY	MATCH
	at-land 00:14:09 - 00:26:03 #<score e q m s b t: 1 0 0 0 0>
	at-land 00:14:09 - 00:22:09 #<score e q m s b t: 5 0 0 0 0>
	at-land 00:00:00 - 00:27:15 #<score e q m s b t: 1 0 0 0 0>
	at-land 00:14:09 - 00:23:20 #<score e q m s b t: 5 0 0 0 0>
	at-land 00:13:05 - 00:59:24 #<score e q m s b t: 3 0 0 0 0>
	rock-hudson 00:20:02 - 00:20:25 #<score e q m s b t: 2 0 0 0 1>

Figure 40. Score Window for the fifth sequence returned

The tenth sequence from the Result Palette offers an interesting three-shot version of “Maya diving off a cliff into the sea and washing up on a beach.”



Figure 41. Stills from the tenth sequence returned

The first shot of the sequence is a portion of the first shot of the best matching sequence in which a character of indeterminate sex dives off a cliff into the sea. It is the part in which the location is the cliff. The second shot is a shot of the sea that happens to work well for two reasons: it has breaking surf that seems to have been created by something falling into the water, and the shot is at an angle and height which imply that it was taken from a cliff above. This sequence does not match as highly as others

because the temporal relation of the action of diving to the location of cliff and then to the location of sea is not satisfied by the sequence: there is no diving taking place in the second shot. Being able to represent and use the knowledge that this is nevertheless a very good match (largely because of what the foaming surf and camera angle of the second shot imply) is an open research problem because of the type of background knowledge a system would have to be able to represent and have access to in order to “understand” how this sequence “works”. The fact that Media Streams was at all able to retrieve and construct this sequence as one of its first ten matches is a testament to the efficacy of its representation and retrieval mechanisms. The reasons this sequence matched are explained in the Score Window below.

#<template #x679FDD9>		
QUERY	MATCH	
		at-land 00:14:09 - 00:26:03 #<score e q m s b t: 1 0 0 0 0 0>
		at-land 00:14:09 - 00:22:09 #<score e q m s b t: 5 0 0 0 0 0>
		at-land 00:00:00 - 00:27:15 #<score e q m s b t: 1 0 0 0 0 0>
		at-land 00:14:09 - 00:23:20 #<score e q m s b t: 5 0 0 0 0 0>
		at-land 00:13:05 - 00:59:24 #<score e q m s b t: 3 0 0 0 0 0>
		rock-hudson 00:24:25 - 00:28:08 #<score e q m s b t: 1 0 0 0 0 0>
		rock-hudson 00:23:20 - 00:24:24 #<score e q m s b t: 1 0 0 0 0 0>

Figure 42. Score Window for the tenth sequence returned

If we modify the original query to ask for a sequence in which Maya’s diving occurs entirely at sea (no cliff), we retrieve a different best sequence and improve the scores of our two “ship diving” sequences.

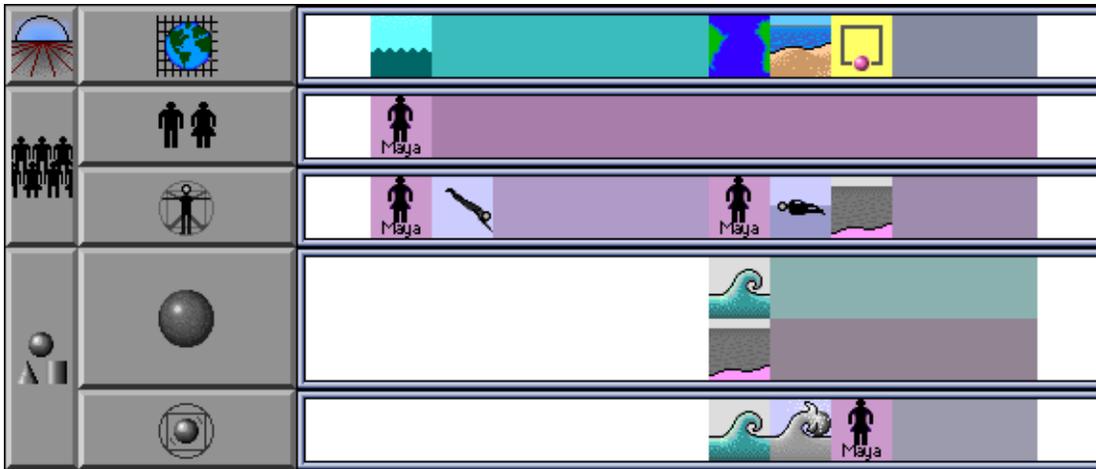


Figure 43. A modified query in which the diving occurs only at sea

In the Result Palette for the above query, we see that the ship diving sequences have moved up to share second place.

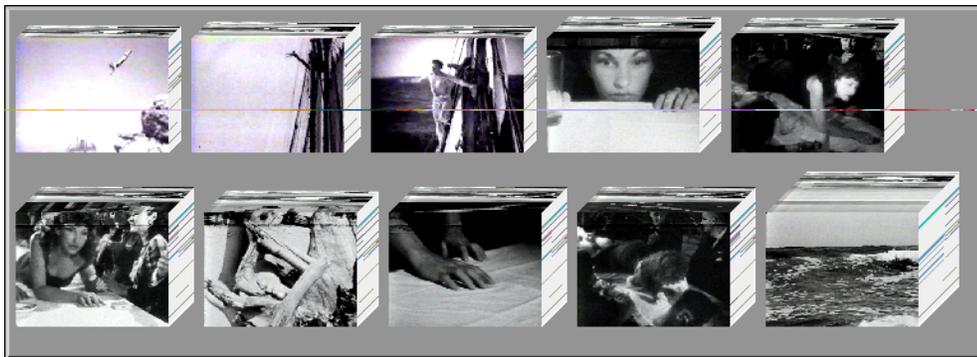


Figure 44. The Result Palette for the query in Figure 43

The best-matching sequence is shown below.

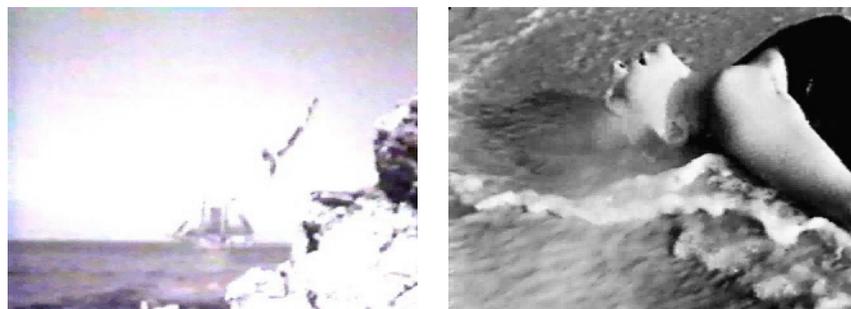


Figure 45. Stills from the best-matching sequence returned by the query in Figure 43

The first shot of the sequence is a portion of the first shot of the former best matching sequence in which a character of indeterminate sex dives off a cliff into the sea. It is the part in which the location is the sea (not the cliff). Media Streams' stream-based representation of video content has allowed the same annotated video stream to supply two different segments of the same original shot in order to satisfy two different queries. The reasons this sequence matched are explained in the Score Window below.

#<template #x66A5DF1>		
QUERY	MATCH	
 Maja		rock-hudson 00:24:25 - 00:29:01 #<score e q m s b t: 2 0 2 0 0 0>
		at-land 00:13:05 - 00:59:24 #<score e q m s b t: 3 0 0 0 0 0>
 Maja		
 Maja	 Maja	at-land 00:14:09 - 00:23:20 #<score e q m s b t: 5 0 0 0 0 0>
		at-land 00:14:09 - 00:26:03 #<score e q m s b t: 1 0 0 0 0 0>
 Maja	 Maja	at-land 00:14:09 - 00:22:09 #<score e q m s b t: 5 0 0 0 0 0>
		at-land 00:00:00 - 00:27:15 #<score e q m s b t: 1 0 0 0 0 0>
		rock-hudson 00:28:09 - 00:32:21 #<score e q m s b t: 1 0 0 0 0 0>

Figure 46. The Score Window for the sequence shown in Figure 44

The two ship-diving sequences are tied for second best matching sequence.

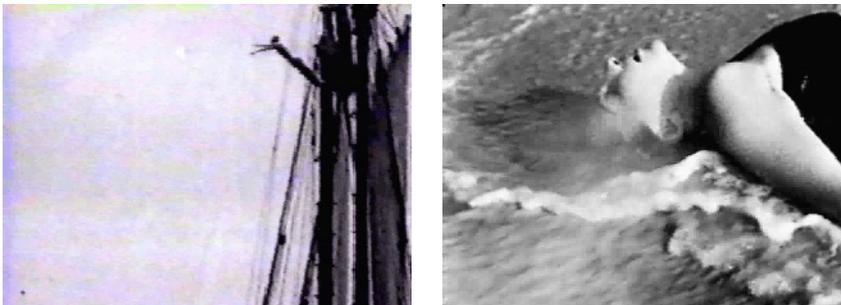


Figure 47 A.

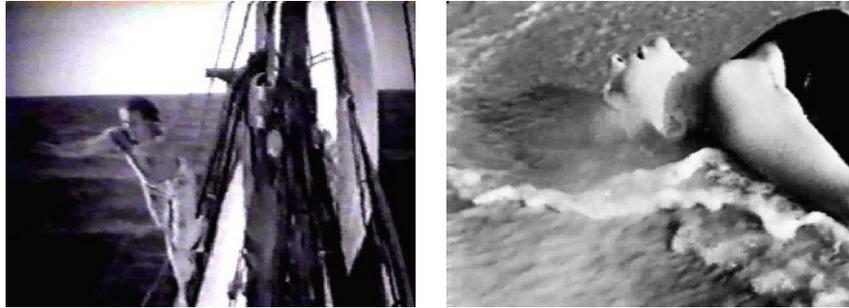


Figure 47 B.

These two sequences now have higher scores since the diving actions of their first shots take place at sea. Media Streams' representation for spatial location helped insure that we could retrieve these sequences when asking for locations at sea, because when creating an annotation of a spatial location the representation encourages that its geographic, functional, and topological features be specified. The reasons these sequences matched (better this time) are explained in the Score Windows below.

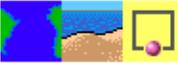
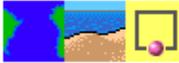
#<template #x66AD741>		
QUERY	MATCH	
 Maya		rock-hudson 00:20:26 - 00:23:03 #<score e q m s b t: 2 0 0 0 1 0>
		at-land 00:13:05 - 00:59:24 #<score e q m s b t: 3 0 0 0 0 0>
 Maya		
 Maya	 Maya	at-land 00:14:09 - 00:23:20 #<score e q m s b t: 5 0 0 0 0 0>
		at-land 00:14:09 - 00:26:03 #<score e q m s b t: 1 0 0 0 0 0>
 Maya	 Maya	at-land 00:14:09 - 00:22:09 #<score e q m s b t: 5 0 0 0 0 0>
		at-land 00:00:00 - 00:27:15 #<score e q m s b t: 1 0 0 0 0 0>
		rock-hudson 00:20:02 - 00:23:03 #<score e q m s b t: 1 0 0 0 0 0>

Figure 48 A. The Score Window for the sequence shown in Figure 47 A

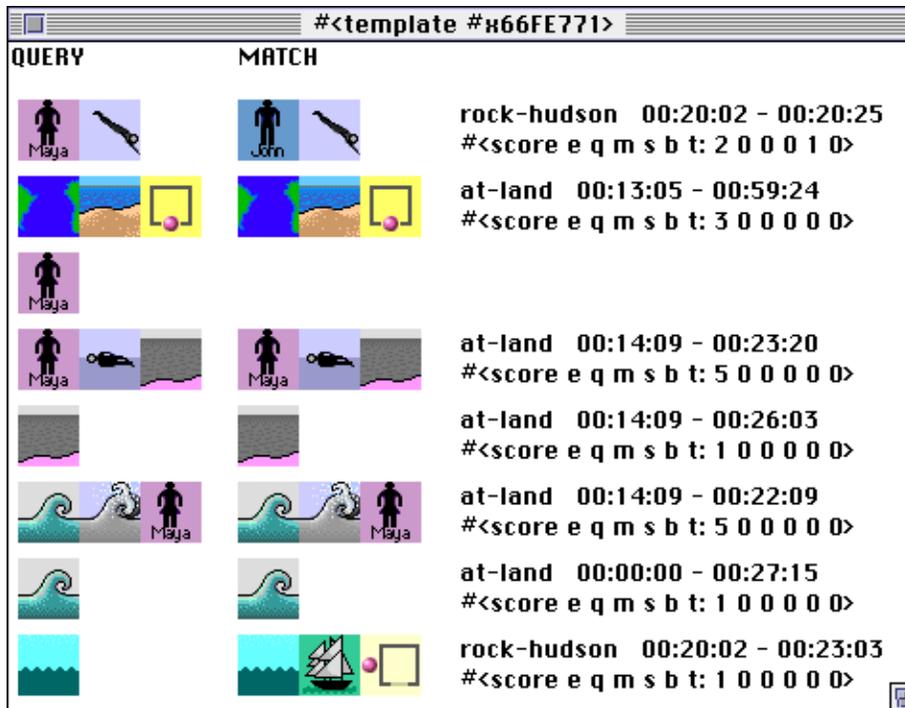


Figure 48 B. The Score Window for the sequence shown in Figure 47 B

If we modify the original query once again this time to ask for a sequence in which Maya's diving occurs entirely at the cliff (no sea), we retrieve a different best sequence.

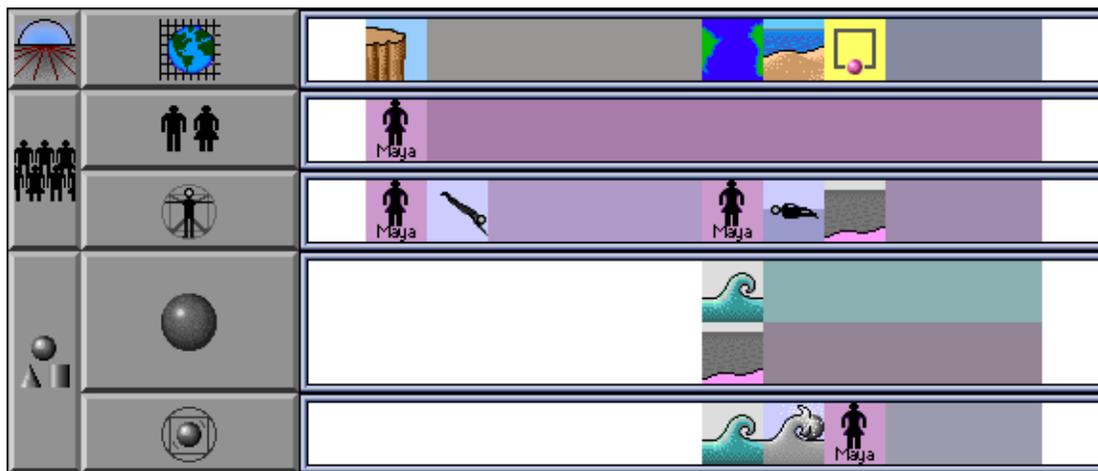


Figure 49. A query in which the diving occurs only at the cliff

In the Result Palette for the above query, we see that the ship diving sequences have moved back to share second through tenth place with equally scored sequences of Maya performing actions of locomotion in various locations.

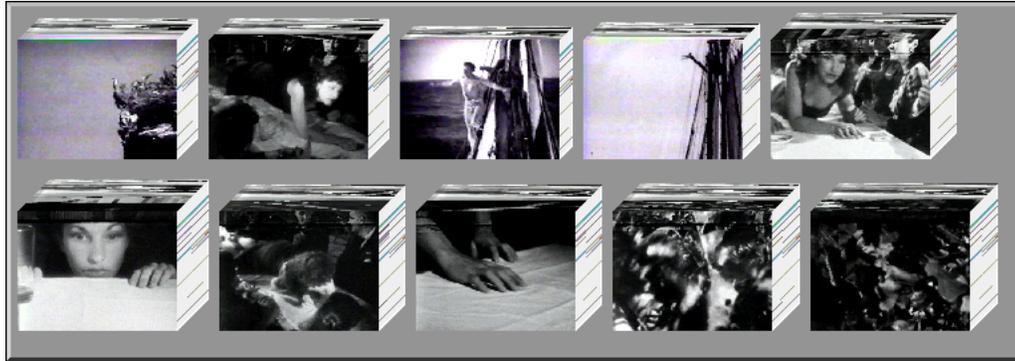


Figure 50. The Result Palette for the query shown in Figure 49

The best matching sequence is shown below.

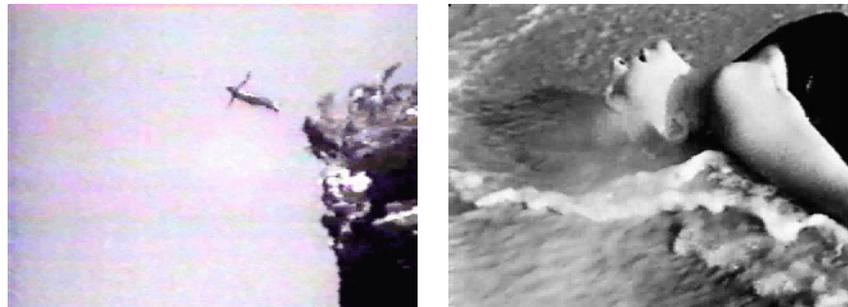


Figure 51. Stills from the best-matching sequence obtained by the query shown in Figure 49

The first shot of the sequence is a portion of the first shot of the former best matching sequence in which a character of indeterminate sex dives off a cliff into the sea. It is the part in which the location is the cliff (not the sea).

What these queries and result sequences show is Media Streams' ability to dynamically *resegment* and *resequence* annotated video in response to user queries according to their semantic, relational, and temporal similarity.

5.3.2. Repurposing and New Forms of Continuity

The practice of making new sequences out of parts of old ones has a long tradition dating back to the origins of cinema. We have already spoken of the experiments of Kuleshov and his students which uncovered the basic mechanisms of montage. Filmmakers from Esfir Shub (Shub 1927) to Bruce Connor (Connor 1958) to M. V. D. (M. V. D. 1990) have developed various aesthetics of repurposing found materials.

Media Streams supports the representation and retrieval of video content so as to make video more repurposable. The aesthetics of repurposed video are not the same as those of traditional Hollywood cinema, but do share some of the same principles of cinematic continuity. *What distinguishes repurposed sequences is that one or a few forms of continuity become dominant enough to override the discontinuity of other aspects of the content.* What follows are the beginnings of a typology of these forms of continuity:

- **Continuity of Actor**
The exploration of the continuity of actor is a common practice in cross-over fan video making (Jenkins 1992) and in innovative forms of documentary and biography such as *Rock Hudson's Home Movies* (Rappaport 1993).
- **Continuity of Role**
The *Archive Films Demo Reel* described in Chapter 4 makes extensive use of this form of continuity (Archive Films 1992).
- **Continuity of Location**
Another of Kuleshov's early experiments was the construction of continuity of location between locations which are discontinuous. He created artificial locations through the careful montage of shots from different locations (Kuleshov 1974: 8). This technique is commonly seen in standard cinema in the use of establishing shots, the combination of location and studio shots, and in shooting almost all television shows, regardless of implied location, in Toronto.

- **Continuity of Action**

This is perhaps the most powerful form of continuity; since action, more than any other aspect of video content, tends to drive the development of video sequences. The continuity of action can be reinforced by the construction of reaction sequences, sequences that use clear temporal chronology, and sequences in which shots are matched on the direction of movement.

These forms of continuity imply a certain type of video comprehension tied to fairly traditional notions of cinematic action and narrative. They are part of the fundamental glue that binds shots together into an intelligible sequence. This level of continuity is not at the level of story continuity but at the level of the continuity of actions, characters, and locations that enables the formation of larger narrative structures. These sub-narrative continuities of video sequences are intimated by Roland Barthes in his discussion of “action sequences” in narrative:

[...] there remain nonetheless within the classical text (before the advent of modernity) a certain number of actional data linked among themselves by a *logico-temporal* order (*this* which follows *that* is also its consequence), organized thereby in individuated series or sequences (for example: 1. to come to a door; 2. to knock at this door; 3. to see someone appear at it), whose internal development (even if imbricated in that of other parallel sequences) affords the story its progress and makes the narrative a processive organism, moving towards its “goal” or its “conclusion.” (Barthes 1971: 138-139)

Barthes develops the beginnings of a taxonomy of action sequences (consecutive, consequential, volitional, reactive, durative, and equipollent) which, in conjunction with Burch’s taxonomy of cinematic transitions (Burch 1969) and Metz’s analysis of the syntagmatic figures of cinematic discourse (Metz 1974), might provide a useful middle ground between the physically-based representation of video in Media Streams and the higher level representations of narrative used in story analysis and generation.

There is also a host of non-narrative forms of continuity which operate along different organizing principles than most video sequences. One such principle is the idea of the match of the graphical appearance of objects between shots that was used extensively in the avant-garde classic, *Ballet mécanique* (Léger and Murphy 1924). Another principle is the creation of a thematic continuity that connects shots in a sequence.

Eisenstein's famous sequence of religious symbols and idols in *October* (Eisenstein 1928) is a well-known example of thematic continuity.

Because of its representational design and retrieval-by-composition methods, Media Streams can retrieve/generate video sequences in response to user queries which did not exist as sequences in the database. These "micro-narratives" (not stories, but the building blocks of stories like Barthes' "action sequences") sequence video segments from various movies and exhibit combinations of the above forms of continuity. In the next section we will look at a sampling of some Media Streams queries and the sequences that were constructed to satisfy them.

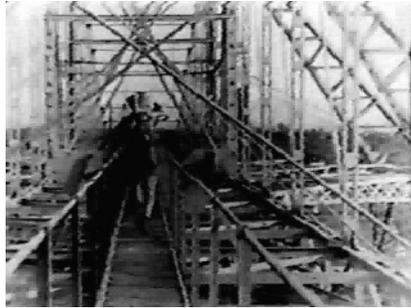
5.3.2.1. Retrieval-By-Composition Examples

In the following examples, all the query results are video sequences whose parts were combined through the retrieval process. Importantly, none of the queries specifies where shot boundaries should occur. The segmentation of the query into shots and the combination of these shots into sequences is accomplished by the system. If Media Streams can find a contiguous segment of video to match a query, it will; if not, it will combine partial matches into a sequence that better satisfies the query. Users can specify where shot boundaries should occur, but as the examples below attest, Media Streams in no way requires that the user formulate the segmentation of the query in order to retrieve/construct a video sequence.

A.

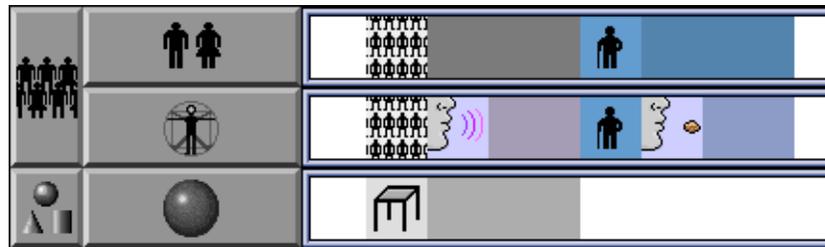


This query is a simple one of a man walking away from the camera and then to the right.



The resulting sequence preserves Continuity of Action (action is satisfied exactly), and synthesizes Continuity of Location (due to the urban setting of both scenes) and Continuity of Actor (due to fact that the face of the character walking away from the camera is not visible and both characters are wearing not radically different clothes).

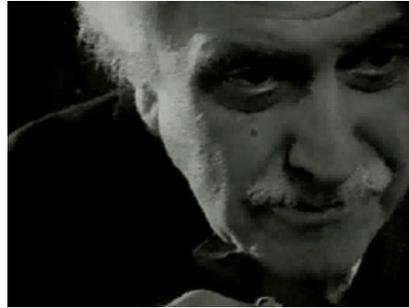
B.



This query is of a crowd making noise with their mouths, a table is present, and then an elderly man is eating.



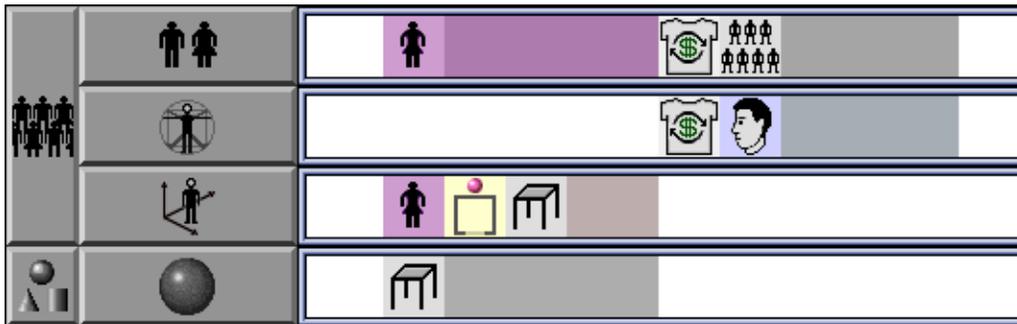
Result 1.



Result 2.

The resulting sequences are Reaction Sequences which by virtue of the semantic contiguity of people around a table and eating synthesize Continuity of Location.

C.



This query is of an adult female behind a table and then seven business people turn their heads.



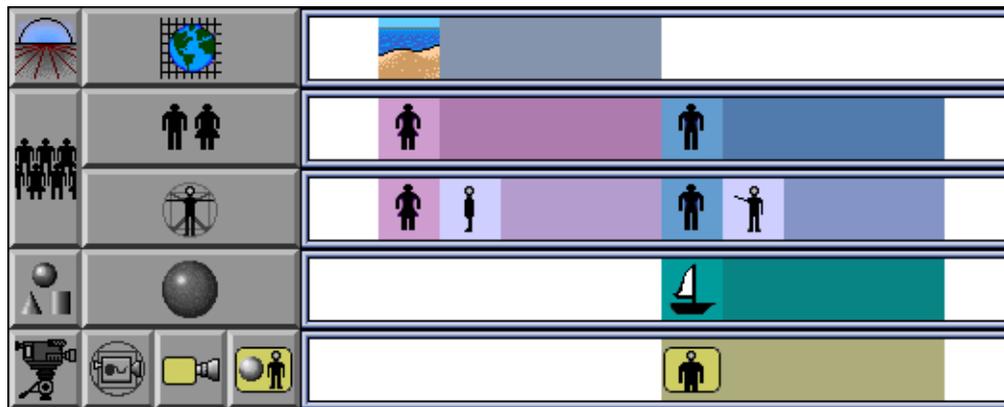
Result 1.



Result 2.

The resulting sequences are also Reaction Sequences (but in a different order than in Query B) which by virtue of the invisibility of the lower half of the bodies of the seven business people synthesize Continuity of Location (being seated at the table which Maya Deren is behind).

D.



This query is of an adult female at a beach rotating her body clockwise and then a medium shot of an adult male waving his right arm with a boat in the shot.



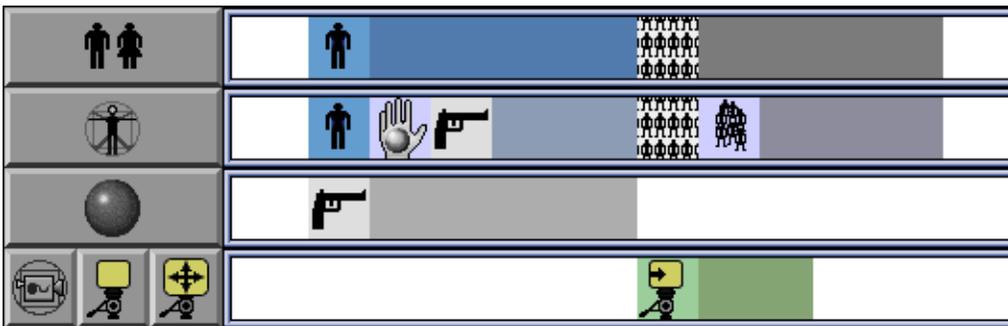
Result 1.



Result 2.

The resulting sequences are both Reaction Sequences which synthesize two different Continuities of Location. In the more exact match, Maya turns from the beach to see John wave to her from a boat. In the less exact match, Maya turns from a beach/desert/sandy place to see Rock wave his gun at her. These two sequences afford different and multiple interpretations of their action sequences. The seeds of many possible stories are contained within them.

E.

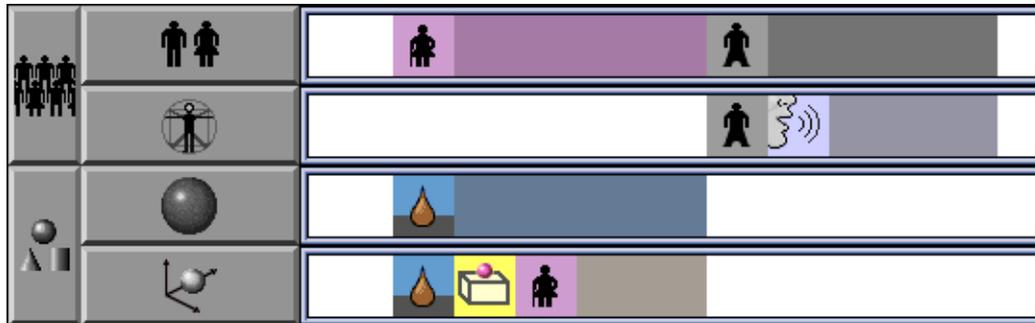


This query is of an adult male using his hand to operate a gun and then a camera pan right shot of a crowd dispersing.



The resulting sequence is a Reaction Sequence that does not succeed in synthesizing Continuity of Location. Nevertheless, it works as a sequence due to the intensity of the action-reaction pair and the resulting Continuity of Action. Even though the action-reaction dynamics of this sequence are coherent, the emotional dynamics are usefully indeterminate: is Rock scaring the crowd or rallying them?

F.



This query is of mud on top of an elderly female and then a character of indeterminate sex laughing.



Like the sequence before it, this resulting sequence is a Reaction Sequence that does not succeed in synthesizing Continuity of Location. Nevertheless, it works as a sequence due to the intensity of the action-reaction pair and the resulting Continuity of Action. The action-reaction dynamics of this sequence are coherent and more constrained than the sequence above since the emotional semantics of a laughter reaction have a somewhat more limited range than the more ambiguous action-reaction pair of raising a gun and a crowd dispersing. We may still wonder whether Rock had foreknowledge of what would befall the elderly woman, or whether he was the cause of her predicament, or whether he simply was an innocent observer of a humorous sequence of events that led up to the first shot in this sequence.

G.



This query is of a character of indeterminate sex emitting smoke with a cigarette in the shot and then an object burning.



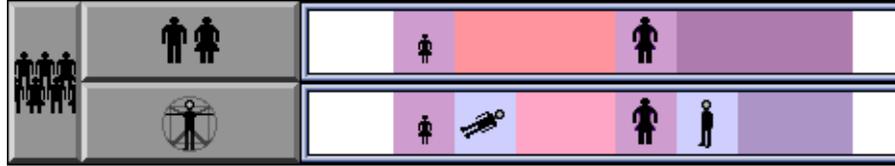
Result 1.



Result 2.

The two resulting sequences are Thematic Sequences whose thematics are reinforced by the Continuity of Action. Media Streams also makes use of its semantic hierarchy to infer that where there is fire there may be smoke. Consequently, if a shot of an object burning does not exist in the database, a shot of an object emitting smoke will be a close match.

H.



This query is of a young female spinning clockwise around a body axis perpendicular to the navel and then an adult female spinning clockwise around a body axis perpendicular to the top of the head.

Result 1.



Result 2.



The two resulting sequences employ multiple forms of continuity which make this micro-narrative have a fairly determinate semantics. The exact Continuity of Action, the apparent Continuity of Actor created by the Chronology of the Sequence as a growth progression from young to adult female, and the exact Match on Direction of Movement in the first sequence (in the second sequence the character is rotating around the same body axis but counter-clockwise) all contribute to making these sequences readable as the story of a happy young girl who grows up and keeps on spinning. The first shot is taken from a home video and the second shot from a Julianna Hatfield music video. The change from classical to rock music in the soundtracks of each shot also adds an interesting dimension to the creation of continuity and narrative in these sequences.

These various examples of retrieval-by-composition and the continuities they employ provide some indications of which aspects of video content may aid in the creation of fragments of narrative activity. Most of the relevant features seem to be connected to the movement of the focus of attention, expectation, and intention as they are made apparent through visual (and auditory) cues: gaze, gesture, motion towards and away from other people and in and out of spaces, and especially, action and reaction. These features are very reminiscent of the cinematic vocabularies and practices developed for the creation and maintenance of narrative action in silent film (Elsaesser 1990; Kuleshov 1974). Further research on creating vocabularies on top of Media Streams' representations which foreground these features would be worthwhile.

5.3.3. Learning from Retrieval

Media Streams also provides a means whereby the system can learn from retrieval. I use "learn" here in a very limited sense. Learning from retrieval means being able to index human-tutored examples of sequence-dependent exceptions to the semantics of the ontology and being able to use these indexed exceptions in later search and retrieval. Media Streams does this by providing a way to *shift* the prototype of an annotation in a retrieved Media Time Line to the prototype of an annotation in the query Media Time Line with which it formerly had no prototype in common, and then to *reindex* this shifted annotation under its new prototype (and its prototypes) in the CIDIS.

In order to understand how this shifting and reindexing works and why it might be useful, let's imagine we've created a query for "adult male eating food." We get our best hit: "Steve Martin eating pizza."



We also get an assortment of other hits like “an elderly male eating food”, “an infant female eating oatmeal”, and quite far down in our results we get “Charlie Chaplin eating a shoe.”



This hit was retrieved because *Charlie Chaplin* matched *adult male*, and *eating* of course matched *eating*, but *shoe* did not match *food*. In the video of Charlie Chaplin eating a shoe, the shoe, which is semantically highly

dissimilar to food, is episodically *functioning as* food. By shifting the prototype of and reindexing this instance of *shoe functioning as food*, we encode this episodic meaning in the Indices of the semantic memory. Once this has been done, a new query for “adult male eating food” would retrieve “Charlie Chaplin eating a shoe” with a higher score because this instance of *shoe* would now be an indexed spinoff of *food*. Media Streams’ ability to learn from retrieval extends the boundaries of categories in the semantic memory by shifting the prototypes of and reindexing context-dependent exceptions to the system’s ontology.

In this chapter we examined how Media Streams’ representations of video content, memory structures, and retrieval algorithms enable the system to retrieve and repurpose video. Media Streams borrows structural and functional principles from human memory in its implementation of semantic and episodic memory structures. We also described our mixed representational system of CIDIS, Media Time Lines, and Indices that support retrieval of video content according to semantic, relational, and temporal similarity. We have shown examples of retrieval-by-composition in Media Streams and some of the types of sequences and forms of continuity it produces. Finally, we have pointed towards how indexing of episodically unique relations and semantics can extend the system’s ontology. In the next chapter, we address the interface issues we had to confront and the solutions we developed in building Media Streams. Interface design is not optional in solving the problem of representing video for retrieval and repurposing. For as we stated in Chapter 4, the task of representation is and will be a very human problem.



Chapter Six

Media Streams Interfaces

6. Media Streams Interfaces

6

In this section we will discuss some of the issues and innovations of Media Streams' interfaces for video annotation, retrieval, and repurposing. Many of the details of the components and functions of the interface are illustrated and discussed in Appendix A. The following discussion will focus on the underlying challenges of designing interfaces for video representation and retrieval, and on some of the specific solutions Media Streams has to contribute.

The syntax and semantics of video tell us what aspects of video content we need to describe in order to facilitate retrieval and repurposing, but tell us little about the interface we should use to describe video. However, the question of representation is tightly linked to the question of interface. Since annotation will be largely a manual process, the representation must be readable and writable by human beings. Any interface for video annotation and retrieval must address two fundamental issues: creating and browsing the space of descriptors to be used in annotation and retrieval; and annotating, browsing, and retrieving video shots and sequences. The interface to the space of the descriptors needs to support:

- reuse of descriptive effort
- visualization of the semantics of the space of descriptors
- creation of new descriptors

An interface for the annotation, browsing, and retrieval of video must contend with the temporality and information density of the medium. Such an interface should support:

- visualization and manipulation of video at multiple timescales simultaneously
- reading and writing of multi-layered annotations
- visualization and manipulation of relations between descriptors

Media Streams' most obvious first answer to the above issues and requirements is the development of an *iconic visual language* for video annotation and retrieval. The functionality and components of the Media Time Line and the Icon Space (with its Icon Workshop and Icon Palettes)

also seek to address the above issues. In order to clarify the issues which confront the design of interfaces for video representation, retrieval, and repurposing, we will divide this chapter into discussions of visualizing, annotating, browsing, and retrieving video.

6.1. Visualizing Temporal Media

6.1.1. Spatializing Time

The history of visual arts and scientific visualization provides many examples of techniques for using space to represent time (Tufte 1990). From ancient Egyptian bas-relief (Groenewegen-Frankfort 1987) to Medieval narrative painting (Gardner and others 1991) to contemporary comic art (McCloud 1993), various techniques have been developed which seek to capture the flow of motion or sequence of events in a two dimensional representation. Depending on which level of temporal representation is the focus, different techniques have been employed. Similar to the distinction between Media Streams' level of physically-based action description and the higher levels of narrative structure which film and literary theory seek to describe, one can differentiate between those visualization techniques which attempt to spatialize *motion* and those which attempt to spatialize *narrative*. A further distinction in techniques for spatializing time is between those which use a *single image* and those which use a *series of images* to represent temporal information.

During and soon after the period of the invention of cinema at the end of the last century, the visual arts were obsessed with exploring spatial representations of motion and temporal events (Kern 1983). The advent of Cubism saw the invention and formalization of two essential techniques for spatializing time (Cooper 1970). Let us consider two fundamental situations in the representation of dynamic events:

- a moving object is viewed by a stationary observer
- a stationary object is viewed by a moving observer

Marcel Duchamp offers us a technique for spatializing time in the case of a moving object viewed by a stationary observer. In his painting, *Nude Descending a Staircase #2* (Duchamp 1912), we see the superimposition of multiple temporal views of a moving object within one image:



Figure 52.

The techniques of stop action photography of horses and people in motion (Bordwell and Thompson 1990: 372; Kern 1983: 21; Musser 1990: 48-54), pioneered by Eadweard Muybridge and Étienne-Jules Marey, can be thought of as the disassembling of Duchamps' representation, or, on the other hand, Duchamp's representation can be thought of as the collapsing of stop action photographs into a single image. Storyboarding techniques resemble Muybridge's and Marey's techniques, but sample the temporal flow of events at irregular intervals in order to convey not motion, but narrative flow.

Robert Delaunay offers us a technique for spatializing time in the case of a stationary object viewed by a moving observer. In his painting, *The Eiffel Tower* (Delaunay 1911), we see the compositing of multiple temporal views made by a moving observer of a stationary object within one image:



Figure 53.

Current techniques for visualizing the temporal structure of video content have much to learn from Cubism and the history of attempts to solve this problem in the visual arts. As we will see below, solutions developed in our work and by other researchers borrow ideas from earlier techniques of spatializing motion and narrative in single and multiple images.

6.1.2. Media Streams Visualization Interfaces

In Media Streams, we make use of single image and image series techniques developed for spatializing motion.

6.1.2.1. Thumbnails

Since video is a temporal medium, the first challenge for representing and annotating its content is to visualize its content and structure. In the Media Time Line we represent video at multiple timescales simultaneously by trading off temporal and spatial resolution in order to visualize both the content and the dynamics of the video data. We create a sequence of thumbnails of the video stream by subsampling the video stream at a rate

of one frame every second. For longer movies, we sample a frame every minute as well.



Figure 54. A stream of “thumbnails,” or subsamples, of a video

The spatial resolution of each thumbnail enables the user to visually inspect its contents. However, the temporal resolution is not as informative in that the sequence is being subsampled at one frame per second. Thumbnails constitute a type of “dumb storyboard” of video content since temporal sampling is at regular as opposed to content-driven intervals. Even “smart storyboards” would suffer the limitation of temporal subsampling which short of sampling almost every frame loses much of the motion information and dynamics of the video content.

6.1.2.2. Videogram

In order to overcome the lack of temporal resolution, we extend a technique pioneered by Ron MacNeil of the Visible Language Workshop of the MIT Media Laboratory (MacNeil 1989; MacNeil 1990; MacNeil 1991a; MacNeil 1991b). We create a *videogram*. A videogram is made by grabbing a center strip from every video frame and concatenating them together. The concatenated strip provides fine temporal resolution of the dynamics of the content while sacrificing spatial resolution. In addition to clearly representing shot boundaries and transitions, a videogram provides a visualization of camera and/or object motion in the video. Because camera operators often strive to leave significant information within the center of the frame, a videogram preserves a salient trace of spatial resolution. In a videogram, a still image has an unusual salience: if a camera pans across a scene and then a center strip is taken from each video frame, a still will be recreated which is coherently deformed by the pace and direction of the camera motion and/or the pace and direction of any moving objects within the frame. Videograms with unchanging strips convey lack of motion through the center of the frame.

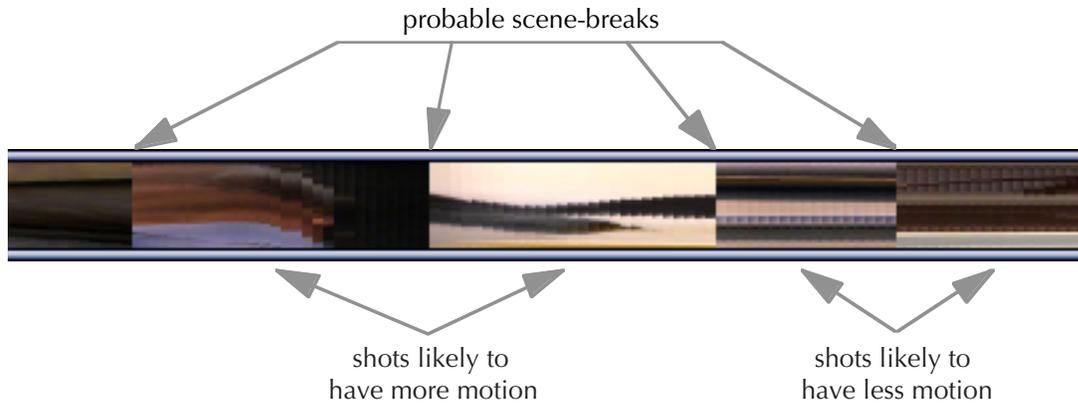


Figure 55.

Our contribution is that by simultaneously presenting two different, but coordinated views of video data—the thumbnails, with good spatial resolution and poor temporal resolution, and the videogram, with poor spatial resolution but good temporal resolution—the system enables the viewer to use both representations simultaneously in order to visualize the structure of the video information.

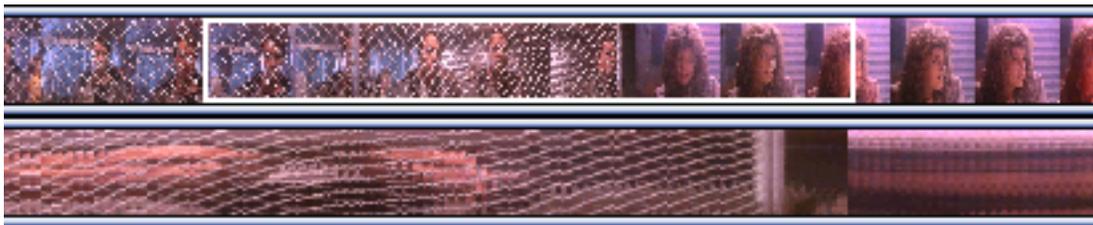


Figure 56.

With little practice, users can learn to read this representation to quickly scan the dynamics of video content from this spatial representation. Shot boundaries are clearly visible as are camera pans, zooms, tracking, and the difference between handheld and tripod recorded video footage. The deformation of the still image in the videogram provides a signature of camera and/or object motion whose content can be interpreted in correlation with the corresponding thumbnails (the thumbnails inside the bounding box are represented by the videogram below).

This idea of playing *spatial* and *temporal* resolutions off one another is also utilized in Laura Teodosio’s work on “salient stills” and holds promise as a general guideline for creating new visualizations of video data (Teodosio 1992). An example of this spatial/temporal tradeoff can be seen in the figure above in which the movement of Arnold through the frame is visible in the right hand side of the videogram. The swath of extended face in the videogram corresponds to the central figure that can be seen in the corresponding thumbnails inside the pale bounding box.

Media Streams’ combination of thumbnails and videograms not only trades off temporal and spatial resolutions, but also visualizes video at multiple timescales.

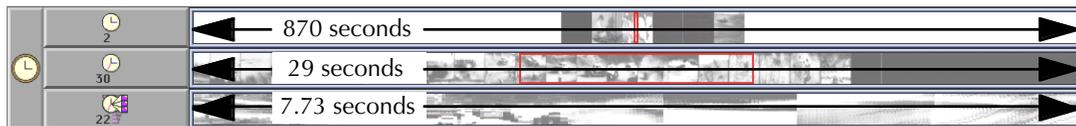


Figure 57.

The idea of providing multiple correlated views of video data is a powerful one for the design of video visualizations. This idea has also been explored in the work of Tonomura (Tonomura and others 1993), Elliott (Elliott 1993), and Smoliar (Zhang and Smoliar 1994). We extend the pioneering work of Mills and Cohen (Mills and others 1992) by combining multi-timescale thumbnail views with a videogram thus creating a correlated, multi-resolution visualization of video data.

6.1.2.3. Waveforms and Pause Bars

Audio data in the Media Time Line is represented by a waveform depicting amplitude, as well as pause bars depicting significant breaks in the audio.



Figure 58.

The visualization of audio content is an open and under-researched problem though some recent work has explored the use of color and scaling (Degen and others 1992).



6.1.2.4. Media Time Line Icons

From picons (single still image), to head-tail pairs (two still images—often the first and last frame of a shot), to micons (multiple moving images—i.e., a movie), to VideoStreamer objects (multiple stacked moving images), designers have tried to develop visualizations of video segments which represent their contents. As mentioned in the previous chapter, Media Streams makes use of Media Time Line Icons to represent video segments and sequences. Media Time Line Icons enable movies to become objects which can be viewed and manipulated within the Icon Space. This visualization is based on the XY-T representation that originated within computer vision. It was adapted by Hirotsada Ueda for representing video segments in his IMPACT system (Ueda and others 1991), and was successfully extended to video browsing and viewing in the recent work of Eddie Elliott (Elliott 1993). Recent work by Tonomura has added another dimension to the XY-T visualization by representing camera pans and zooms through offsetting and/or resizing the video frames. He calls these objects VideoSpacelcons (Tonomura and others 1993).

We utilize and extend this XY-T visualization technique in Media Streams for representing annotated video content. The length of each video segment is indicated by the depth of the volume of the media time line icon.

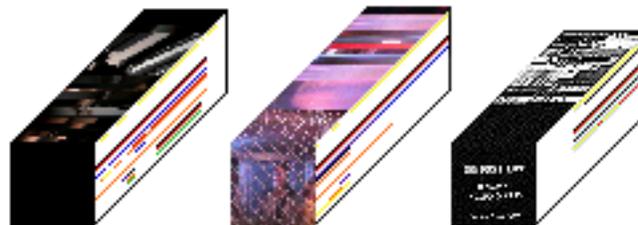


Figure 59.

In order to accommodate the simultaneous viewing of movies of differing lengths, we use *logarithmic scaling* of the volume depth, so that the longest movie and the shortest movie can be seen within the same window without sacrificing the ability to see the relative lengths of all movie segments. We also represent the *density of annotation* by re-presenting the annotations of the Media Time Line as a series of color bars on the side face of the Media Time Line Icon.

6.2. Annotation: Adding Structure

As powerful and useful as the above visualization techniques are, they do not provide a sufficient representation of video for content-based retrieval and repurposing. For that, annotation is required. Annotation is an interface intensive task. In video annotation the problems and complexities of creating a representation for video become especially pressing: the potential tedium of the annotation process functions as an acid test for the robustness, intuitiveness, and efficiency of any possible video representation or interface.

Annotation can be divided into four different tasks (Weber 1994):

- real-time annotation during recording
- non-real-time annotation during recording
- real-time annotation after recording
- non-real-time annotation after recording

Media Streams is a system designed for the fourth type of annotation (non-real-time annotation after recording). Because the process of annotating video in Media Streams' is non-real-time and not tied to the act of recording, one has the ability to refine, review, and redo annotations. As such, it has affinity with notational systems designed for temporal media which served as composition technologies.

6.2.1. Precursor Notational Systems

There are two extant, highly specialized notational systems which deserve special attention in relation to the project of developing an annotation system for video: music notation and movement notation. There are also two practices in filmmaking which have developed less well-defined notational systems for content: storyboarding and film scoring.

6.2.1.1. Music Notation

Music notation spatializes temporal events into discrete spatial marks which can be surveyed and analyzed in ways not possible with the raw temporal events. Modern music notation also provides a coherent framework for visualizing temporal events: the staff. The musical staff was invented in the Medieval period and solved the problems of the underdetermined semantics of prior notational systems, which lacked a

way of clearly demarcating the pitch differences between notes (Read 1969).

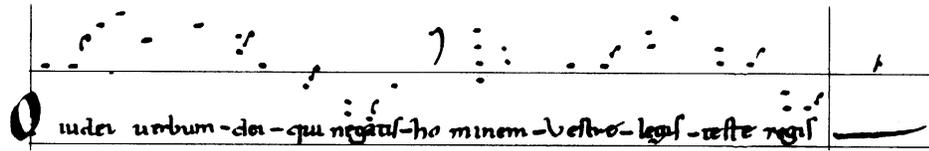


Figure 60. Early Medieval music without a staff



Figure 61. Modern music notation with staves and two instrument parts

Music notation also enables the clear separation of parts for different instruments. This ability is essential for the study and practice of music. Even with the advent of audio recording technology, sheet music is still the preferred way of learning a piece of music and of visualizing its structure. All current time line representations used in video seem derivative of this aspect of music notation. Because music and video notation are both concerned with representing complex temporal events, we borrow from music the time line notation, which enables a multi-stream notation of events over time. One can see the annotation streams of the Media Time Line as a type of staff that separates the various annotations into staves of clearly demarcated semantic categories.

Modern music has revealed some limitations in traditional music notation. Some composers have needed to express durations in seconds, as opposed to measures and beats. In this regard, MIDI scores may be seen as an advancement in the temporal precision and expressivity of music notation systems.

As opposed to audio recording, music notation also introduces the difference between the score and its various performances. An audio recording is always a particular performance. A score is a structure that can be used to create many performances. The analogy to video is an interesting one. The use of video notation for storyboarding captures the relationship of score to performance: the storyboard is the score; the movie is the performance. But the analogy when applied to the retrieval process

creates an interesting distinction. If the query is the score, then the retrieved sequences are performances. Akin to the score-performance relationship, query has a one-to-many relationship with retrieved sequences. In this way, Media Streams' video annotations can be thought of as potential scores for performing new sequences through the retrieval-by-composition process.

6.2.1.2. Movement Notation

Efforts to notate the movement of human bodies in space over time have resulted in over 85 forms of dance notation (Guest 1984). Guest defines dance notation as follows:

Dance notation is the translation of four-dimensional movements (time being the fourth dimension) into signs written on two-dimensional paper. (Note: a fifth 'dimension' — dynamics — should also be considered as an integral part, though usually it is not.

Dance notation is (or should be) to dance what music notation is to music and the written word is to drama. (Guest 1984: xiv).

For example, we can look at a passage of Friedrich Albert Zorn's dance notation from 1887:

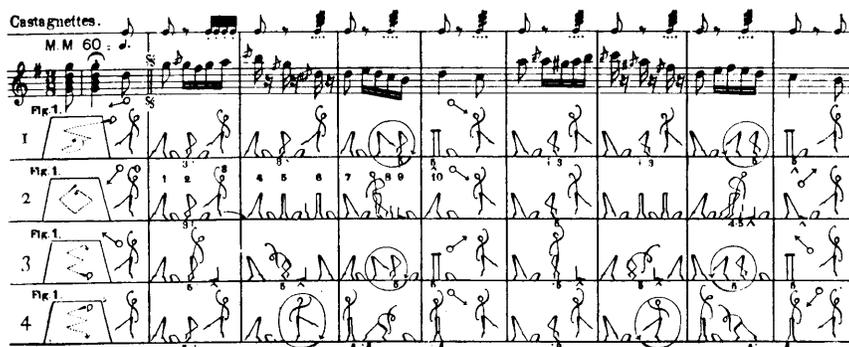


Figure 62. Zorn's dance notation (from Guest)

Dance notations are a subset of the larger set of movement notation systems:

Though dance has been the field which has felt the greatest need of a system of notation, we need, in fact, to look at the wider field of *movement notation*, the recording of any kind of movement, be it gymnastics, sports, therapeutic exercises, anthropological studies, or

the actions of men and objects in outer space. (Guest 1984: xiii).

Guest's survey of the history and classification of dance notation systems is too detailed to cover here. Her comments on the reasons why natural languages are not useful for notating human movement are especially relevant to video notation and highlight some of the advantages of using a designed visual language for notation:

Any serious system of movement notation avoids words because they are also a deterrent in international communication. Then there is the practical consideration that a symbol is briefer and can be read more swiftly than words. [...]

One of the advantages of a notation system is organization, the placement of information where it is easily located. There is no universal standardization as to where specific pieces of information will be placed in word descriptions. (Guest 1984: 14).

Interestingly, even with the advent of affordable video recording technology, many dancers still use dance notations to learn dances. As sheet music for musicians, dance notation captures and makes legible the salient aspects of the temporal stream of events, and makes possible the separation of score and performance.

6.2.1.3. Storyboarding

The mélange of storyboarding techniques do not constitute a notional system as such but represent a series of idiosyncratic practices of articulating the structure and content of a movie. One can glean several useful tools from these various practices: a strategy for capturing salient temporal samples of the visual data; various techniques of overlaying multiple content representations and notations (camera motion, camera angle, direction of object motion, etc.); and often compact and efficient textual representations of action (Katz 1991).

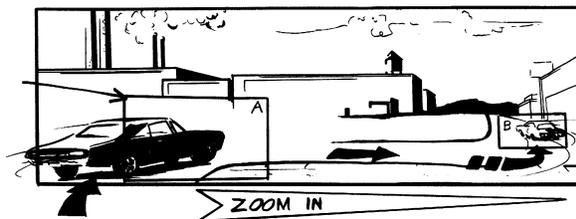


Figure 63.

Advanced storyboards begin to resemble and make use of many of the representational strategies of comic art, especially those for depicting motion and those which use the position, size, shape, and other features of the image frame to convey information (McCloud 1993).

6.2.1.4. Film Scoring

The practice of film scoring, like storyboarding, has a similar degree of idiosyncrasy and variety of practice and technique (Gorbman 1987; Jones 1946; London 1936; Prendergast 1977). In the silent era, film scores were literally *scores* for live musical and acoustical performances meant to accompany the motion pictures (Hofmann 1970). As a compositional notation, film scoring must grapple with the synchronization of visual and auditory streams and the representation of those aspects of the content of each which are salient to their interacting composition. Eisenstein attempted to address this problem of visualizing and creating notations for the content of film and music. His score for *Alexander Nevsky* is informative in this regard, and though I was unaware of it while designing Media Streams, it shares many essential features with our Media Time Line (Eisenstein 1947):

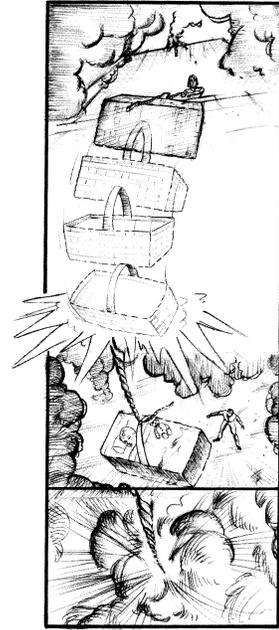


Figure 64.

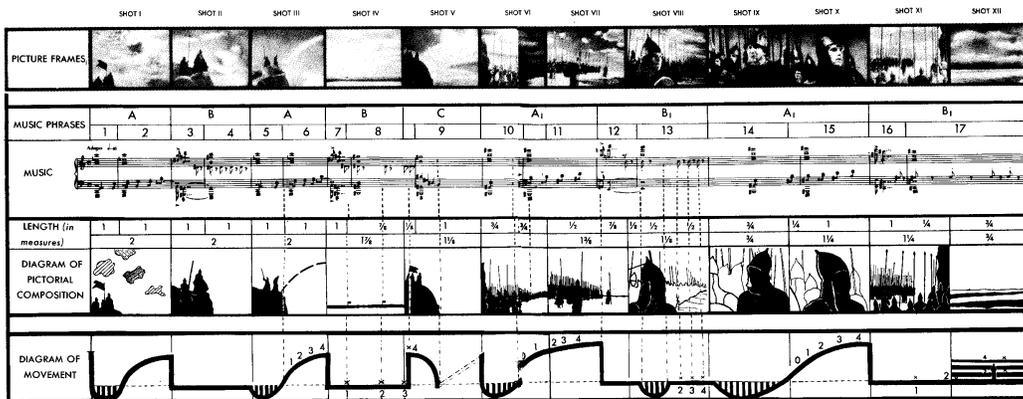


Figure 65.

Eisenstein clearly would have *loved* multimedia computers. As we can see from his diagram, the idea of layering multiple representations of video content which combine visualization with transcription and notation is a powerful idea in the design of interfaces for video annotation.

6.2.2. Annotation at Production Time

Currently, most all information produced at the time of recording video and audio remains inaccessible to annotators and is often discarded after its production. Storyboards, production stills, continuity logs, editing logs, on location comments and annotations all help to record the semantic content of video and audio information. In addition, the physical situation of the recording devices themselves within their production environment can provide useful information about the scene being recorded. A "data-camera" that encoded along with the video signal a time stamp, spatial coordinates, and semantic content annotations recorded on location could pre-log information useful in parsing the content of video and audio (Davenport and others 1991). Computerized systems for organizing and centrally recording production information would not only facilitate the production process (Lasky 1990), but could be extended to help automate the annotation process as well. Future efforts of Silicon Graphics' new Silicon Studio division may point in this direction.

The simple fact is that the earlier in the production pipeline annotation can begin, the better. If annotation can be done automatically or semi-automatically in a form that is usable by machines and humans, then the effort of annotation after recording will be greatly diminished. The possibility of extending Media Streams to support real-time and non real-time annotation during (or before) recording seems plausible, especially through the incorporation of iconic annotation into the recording device or into a handheld logging peripheral.

6.2.3. Making Descriptors in Media Streams

In the last chapter, we discussed Media Streams' underlying representations for video content (CIDIS, Media Time Lines, and Indices). In the following sections, we will describe the interfaces we have developed for creating, finding, and using these representations. The interface to Media Streams' representations is a structured iconic visual language of over 3500 composable descriptors. We create and search for icons in the Icon Space and use them to describe video on Media Time Lines.

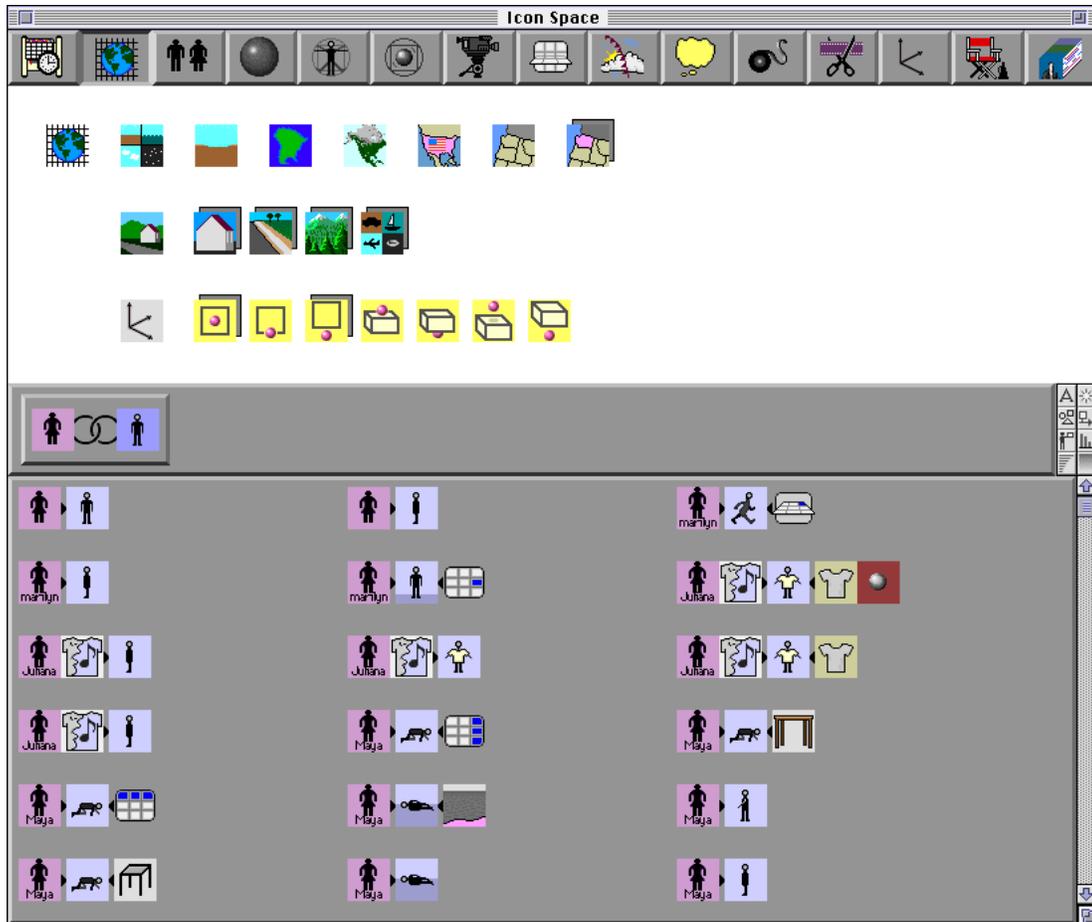


Figure 66. The Icon Space.

The Icon Space is divided into two main sections. The Icon Workshop (the upper half) is where we create compound icons. The Icon Palette (the lower half) is where we search for compound icons that we or other users have already created.

6.2.3.1. Icon Workshop

What enables the user to navigate and make use of our large number of primitives is the way the Icon Workshop organizes icons into cascading hierarchies. We refer to the iconic primitives in the Icon Workshop as *cascading icons*. The Icon Workshop has two significant forms of organization for managing navigational and descriptive complexity:

- **Cascading Hierarchy with Increasing Specificity of Primitives on Subordinate Levels**

Cascading icons are organized in hierarchies from levels of generality to increasing levels of specificity. Similarly to cascading menus on the Macintosh, when a user cascades down an icon hierarchy by clicking on a cascading icon, its subordinate icons are displayed to the right of the cascading icon. These subordinate icons are arranged *horizontally* and represent an increased level of specificity. Some of the icon hierarchies cascade to as many as 7 or 8 levels deep, yet, similarly to the semantic hierarchies of the CYC Project (Lenat and Guha 1990), the design of the categories themselves and their first two or three levels is the hardest and most important representational task.

- **Compounding of Hierarchically Organized Primitives Across Multiple Axes of Description**

In many icon hierarchies in the Icon Workshop, there exists an additional form of organization. When subordinate icons are arranged *vertically*, they represent independent axes of description whose icon hierarchies can be cascaded through separately and whose respective subordinate icons can be compounded together across these axes to form compound iconic descriptors. This form of organization enables a relatively small set of primitives to be compounded into a very large and rich set of descriptors.

To illustrate these forms of organization in our iconic language we can look at how the compound icon for “the scene is located on top of a street

in Texas,” , was created. The figure below shows the cascading icon hierarchy for **spatial location** extended out to the icons for **Texas, street, and on top of**, which the user compounded to create the icon for “the scene is located on top of a street in Texas.”

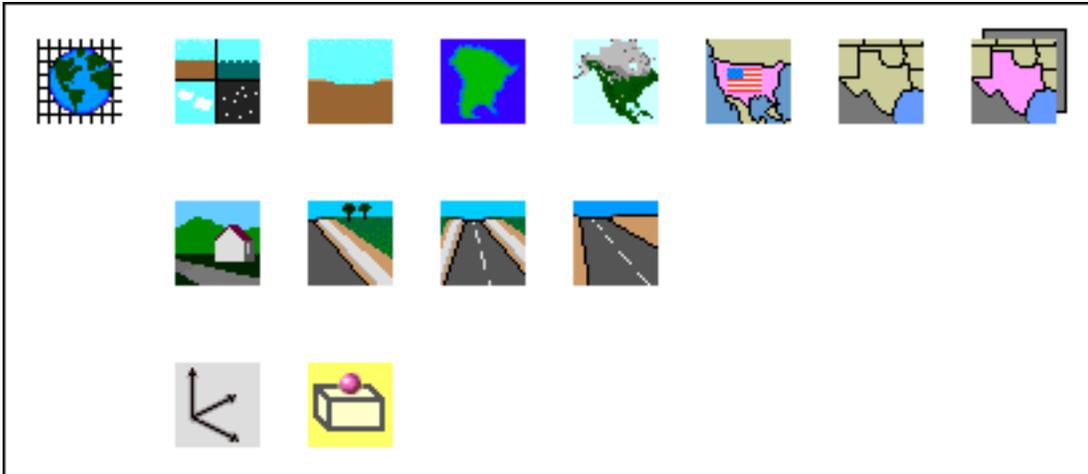


Figure 67. An Icon Path to *On Top of a Street in Texas*

The user clicked on the **spatial location** icon, which cascaded to show its *vertically* arranged subordinate icons **geographical space**, **functional space**, and **topological space**. Each of these cascading icons has further *horizontally* arranged subordinate icons each of which may go several levels deep. For example, the icons in the path from **geographical space** to **Texas** each represents a distinct level of progressive specification (geographical space → land → continent → North America → United States of America → Southern Mid-Western States → Texas). As indicated by the gray square behind the **Texas** icon, it too has further levels of specificity below it which can be displayed by clicking on the icon. In the Icon Workshop, at all but the terminal levels in the hierarchy, there exist many icons which themselves have further levels of specification. At any level in the hierarchy, icons can be compounded across the vertical organization to create compound icons. In addition to clicking, cascading icons can be accessed by voice (using the Voice Navigator II™), by typing in text for their names, or by dropping an existing icon onto the Icon Workshop that opens the icon hierarchies up to the terminals of the components of the dropped icon. In all these ways, a vast, but structured space of icons can be easily navigated by the user.

It is also important to note that the icon hierarchy of the Icon Workshop has like-named nodes in multiple branches. The same iconic primitives can often be reached by multiple paths. The system knows the paths users take to get to these primitives; this enriches the representation of the compounds which are constructed out of these primitives. Having multiple paths allows different categorization schemes to coexist in the Icon Workshop. This is especially useful in the organization of object icons, in which, for example, the icon for **blow-dryer** may be reached under **hand-held device**, **heat-producing device**, or **personal device**.

Cascading icons can be compounded into compound icons in two ways:

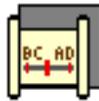
- by *clicking* on one of the desired components
- by *dragging* one of the desired components out of the Icon Workshop

While holding the **⌘** key, if the cursor is over any of the cascading icons in the Icon Workshop, a special  cursor will appear indicating that Media Streams is prepared to compound an icon. Once a compound icon has been created, it can be used immediately to annotate video on the Media Time Line or stored in the Icon Space for use in later annotation or search.

In our discussion of categories for video representation we examined parts of Media Streams' ontology. The Icon Workshop provides an interface to those categories and adds a few others. In the sections below we highlight the *vertical* and/or *horizontal* organization of various Icon Categories in the Icon Workshop that enables users to create compound icons within Media Streams.

6.2.3.1.1. Icon Categories

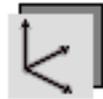
6.2.3.1.1.1. Mise-En-Scene: Time, Space, and Weather



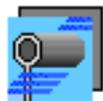
Temporal Location is subdivided *vertically* into historical period (from the age of the dinosaurs through the twentieth century on into the future), time of year (spring, summer, fall, and winter), and time of day or night (morning, afternoon, sunset, midnight, etc.).



Spatial Location is subdivided *vertically* into geographical space (land, sea, air, and outer space), functional space (buildings, public outdoor spaces, wilderness, and vehicles), and topological space (inside, outside, above, behind, underneath, etc.).



Weather is subdivided *vertically* into moisture (clear, partly sunny, partly cloudy, overcast, rainy, and snowy) and wind (no wind, slight wind, moderate wind, and heavy wind). Temperature is not something that can be directly seen; a video of a cold clear day may look exactly like a video of a hot clear day. It is the presence of snow or ice that indirectly indicates the temperature.



6.2.3.1.1.2. Characters and Objects

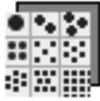


Characters are subdivided *vertically* into characters (female, male, indeterminate sex, and non-human), occupations, which conflate costume and role (personal care, commercial, institutional, religious, sports) and number (one, two, three...many).





Objects are subdivided *vertically* into various types of objects and number of objects.



There are numerous subcategories of objects, many of which mirror subcategories of occupations in the characters hierarchy, and subordinate icons for number (one, two, three, four, five, six, seven, eight, many).

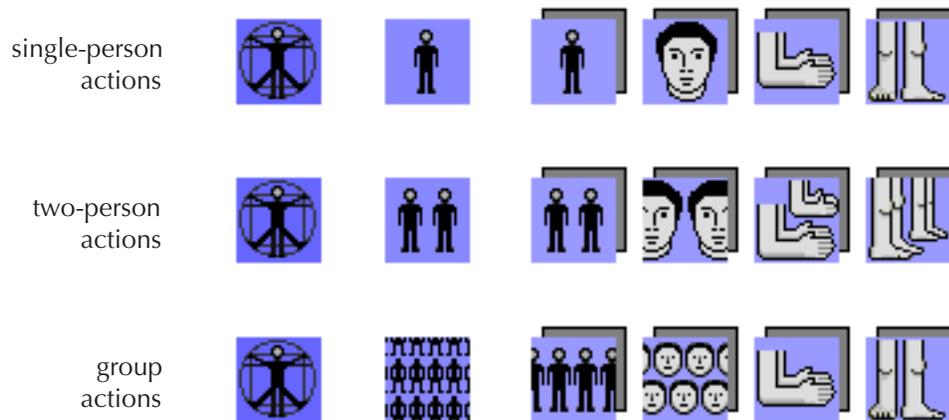


Figure 68. The subcategories of the Objects hierarchy.

6.2.3.1.1.3. Character Actions and Object Actions



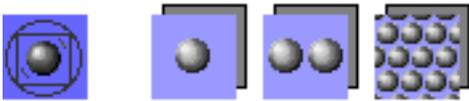
Character Actions are subdivided *horizontally* into single person, two person, and group actions, and each of these is further subdivided *horizontally* into full body actions, head actions, arm actions, and leg actions.



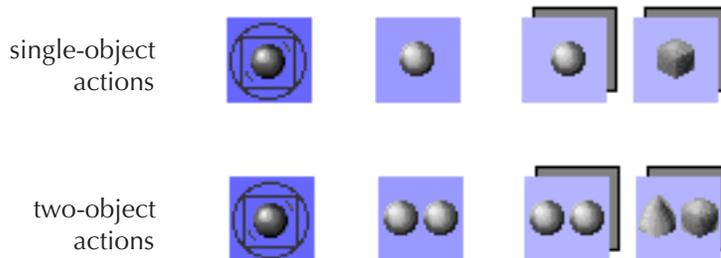
Under each of these categories of human action (and their own subdivisions) action is represented in two ways:

- conventionalized physical motions
- abstract physical motions

We built into our ontology many commonly occurring, complex patterns of human motion which seem to have cross-cultural importance (e.g., walking, sitting, eating, talking, etc.). We also provide a hierarchical decomposition of the possible motions of the human body according to articulations and rotations of joints. Since Media Streams enables multi-layered annotation, any pattern of human motion can be described with precision by layering temporally-indexed descriptions of the motion of various human body parts.



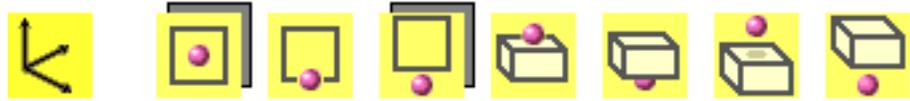
Object Actions are subdivided *horizontally* into actions involving a single object, two objects, or groups of objects. Each of these is divided according to object *motions* and object *state changes*.



For example, the action of a ball rolling is an object motion; the action of a ball burning is an object state change.

We represent actions for characters and objects separately in the Icon Workshop because of the unique actions afforded by the human form. Our icons for various actions are *animated* which takes advantage of the affordances of iconography in the computer medium as opposed to those of traditional graphic arts.

6.2.3.1.1.4. Relative Positions



Relative Positions are used to describe the spatial relationship between characters and objects and are subdivided *horizontally* into inside, on the threshold of, outside, on top of, underneath, above, and below.

6.2.3.1.1.5. Screen Positions



Screen Positions are subdivided *horizontally* into two-dimensional screen position and screen depth.

6.2.3.1.1.6. Cinematography



Cinematography icons are subdivided *horizontally* into “lens” actions (framing, focus, exposure), “tripod” actions (angle, canting, motion), and “truck” actions (height and motion). By layering these iconic descriptors on the Media Time Line, the user can describe simple to very complex camera motions.

6.2.3.1.1.7. Recording Medium



In addition to representing the motions and states of the recording device we also can represent the “look” of the recording medium. Icons for **Recording Medium** are subdivided *vertically* into stock (70 mm film, 8mm video, etc.), color quality (color, black & white, sepia, etc.), and graininess (fine, medium, coarse, etc.).

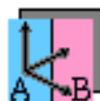
6.2.3.1.1.8. Thoughts



In the early design phases of Media Streams, we spoke with archivists from Monitor Television in Boston about their work practice. They told us that producers would come to them with queries for footage, saying: “Get me something with a lot of action in it!” Or, regarding the framing of a shot: “I want a well composed shot of three Japanese kids sitting on some steps in Tokyo.” These *explicitly subjective* assessments about the qualities of video are addressed in our representation by **Thoughts** icons which are subdivided *vertically* into thoughts about the screen (framing, activity, color) and evaluation (from three thumbs up to three thumbs down).

Thoughts icons, as do all Media Streams icons, allow subjective comments to be typed into a text field associated with the icon. The advantage of still using an icon for subjective comments is that it explicitly articulates the consensual common ground that the comment may address, such as the framing, color, or level of activity in a piece of video as well as the overall gist of the assessment (thumbs up or thumbs down).

6.2.3.1.1.9. Transitions



In the Icon Workshop, we horizontally subdivide **Transitions** between shots according to temporal transitions, spatial transitions, and visual transitions (cuts, wipes, fades, etc.).

When a transition icon is dropped on the Media Time Line, Media Streams creates a compound icon in which the first icon is an icon-sized (32 x 32 pixels, 24 bits deep) QuickTime Movie containing the first shot, the second icon is the transition icon, and the third icon is an icon-sized QuickTime Movie containing the shot after the transition. If we return to our example of the two-shot elevator sequence from Chapter 4:

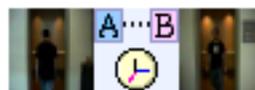


Shot 1: person enters elevator,
elevator doors close

Shot 2: elevator doors open,
person exits elevator

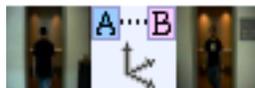
Figure 69. The Two-Shot Elevator Sequence

Its transition icons would be as follows:



Temporal Transition

(forward temporal ellipsis of a determinate length)



Spatial Transition

(spatial translation of a determinate proximity)



Visual Transition
(simple cut with no duration)

We hope to eventually use transition icons to improve Media Streams' knowledge about the world and to facilitate new forms of analogical retrieval. A search using the icons above would enable the user to find a "matching" shot in the following way. The user could begin with a shot of a person getting into an automobile and use one or more of the transition icons as analogical search guides in order to retrieve a shot of the person exiting the automobile in a nearby location. The query would have expressed the idea of "find me a Shot B that has similar transition relations to Shot A as Shot D has to Shot C."

6.2.3.2. Extensibility of the Icon Language

Currently, we have two ways of extending the iconic visual language of Media Streams beyond the composition of iconic primitives. Icons and the components of compound icons can be titled in the **Icon Title Editor** of the **Icon Information Editor**, and new animated icons can be created out of parts of existing ones in the Animated Icon Editor.



Figure 70. The Icon Information Editor and the Icon Title Editor

The Icon Title Editor enables the user to attain a greater level of specificity of representation while still making use of the generality and abstraction of icons. For example, if the user were to annotate video of an automobile with the descriptor "XJ7," this description may be very opaque. If, however, the user titles a car icon "XJ7," in addition to the computer learning that XJ7 is a type of car, a human reading this annotation can see simply and quickly the similarity between an XJ7 and other types of automobiles. A form of system maintenance would be to periodically find titles for which there are many occurrences and create an icon for them.

Users can also create new icons for character and object actions by means of the **Animated Icon Editor**.

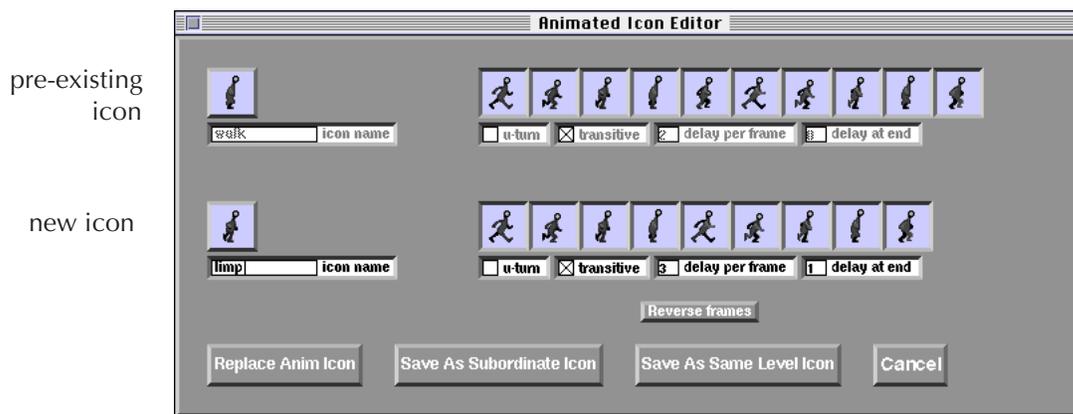


Figure 71. The Animated Icon Editor

The Animated Icon Editor allows users to define new icons as subsets or mixtures of existing animated icons. This is very useful in conjunction with our icons for abstract physical actions, because a very wide range of possible human motions can be described in terms of motions around various joints of the body.

Other possible forms of extensibility for the icon language could attempt to integrate work on automatic icon incorporation that tries to intelligently composite icon parts to form new single icon primitives (Fuji and Korfhage 1991). In our icon language, there are many iconic descriptors which we designed using the principle of incorporation (by which individual iconic elements are combined to form new single icons). Creating tools to allow users to automatically extend the language in this way is a logical extension of our work in this area.

6.2.4. Finding Descriptors in Media Streams

After compound icons have been made in the Icon Workshop they are available for use in the Icon Palette of the Icon Space. Media Streams' Icon Palettes enable users to reuse the descriptive effort of others. If an annotator has created icons for a piece of footage, those icons are available to future users. In this way descriptive effort is reduced and descriptor converge is increased.

6.2.4.1. Icon Palettes

The Icon Palette is the place in the Icon Space where already created compound icons can be retrieved and grouped into palettes on-the-fly. The Query Bar at the top of the Icon Palette is where users can drop down compound icons to form icon queries which retrieve related groups of icons. We have developed an Icon Query Language that enables users to make simple yet powerful queries into the space of descriptors in order to create customized icon palettes on demand.

6.2.4.1.1. Icon Query Language

The Icon Query Language enables users to retrieve compound and glommed icons that are semantically and temporally similar. The idea of creating a palette of icons is to reuse the descriptive effort of others and to create groups of related descriptors. In the Icon Query Language, compound icons can be related to each other in several ways:

- by sharing icons
- by being prototypes or spinoffs of each other (prototype network)
- by being subordinate or superordinate icons of each other (annotation hierarchy)
- by temporally overlapping on Media Time Lines

The Icon Query Language enables users to query for icons as well as for video segments. The reason one might want to query for a video segment in the Icon Query Language is that annotated video segments are excellent repositories of *related sets of descriptors*. If I want to log a treaty signing and I have a video segment of a treaty signing already logged, if I can retrieve its icons, I will already have many of the descriptors I need

grouped together (treaty, diplomats, table, pen, news reporters, writing actions, etc.).

The two basic interface components in the Icon Query Language are *filter units* and *linkers*. Filter units group query elements much like parentheses. This filter unit groups the icons for objects and characters:



It would return all the compound and glommed icons with objects in them and then all the compound and glommed icons with characters in them.

Linkers express relations between filter units and the query elements they contain. There are two linkers:

a  linker (“and”)

a  linker (“temporally overlapping”)

The Icon Query Language interprets the absence of a linker as an “or”. In order to understand the semantics of linkers and filter units, let’s look at some example queries:



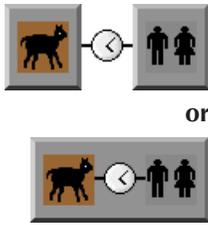
Query for all compound or glommed icons containing components of type X. This instance queries for all compound or glommed icons that contain an object.



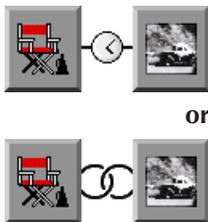
Query for all compound or glommed icons containing components of type X, **or** components of type Y. These instances query for all compound or glommed icons that contain either objects **or** characters. Were the filter units reversed, the results of the query would be presented in the opposite order.



Query for all glommed icons which contain components of types X **and** Y. In this case, these instances query for all glommed icons that contain both an object **and** a character. The order of filter units in an “and” expression is unimportant.



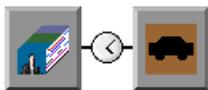
Query for all compound or glommed icons of annotations containing components of type X which **temporally overlap** on a Media Time Line with annotations containing components of type Y. The queries shown here would return the compound or glommed icons of all "land animals" that appear on Media Time Lines at the same time as a character annotation. The order of filter units in a "temporally overlapping" expression is important: to query for all character icons which temporally overlap land animals icons, the filter units would have to be reversed. The first element of a "temporally overlapping" query determines the type of icons returned.



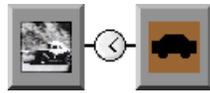
Query for all the compound or glommed icons which **temporally overlap** with the annotations of Movie A. The icon in the left-hand filter unit is a special character which stands for the set of all icons. This query would return all the icons used to describe the Movie represented by the Micon in the right-hand filter unit.



Query for all the compound or glommed icons containing components of type X which **temporally overlap** with the annotations of Movie A. This query would return all the character compound or glommed icons used to describe the Movie represented by the Micon in the right-hand filter unit.



Query for all of the Movies annotated by compound or glommed icons containing components of type X. The icon in the left-hand filter unit is a special icon that stands for the set of all annotated Movies. This query would return, in the form of Media Time Line Icons, all of the segments of Movies which at one time or another are annotated by a "land vehicle."



Query for all of the **segments** of Movie A which are annotated by compound or glommed icons containing components of type X. When a particular Movie is filtered in this manner, the results are Media Time Line Icons which represent the segments of the Movie which satisfy the query. The query shown would return, in the form of Media Time Line Icons, all of the **segments** of Movie A which are annotated by a "land vehicle."



Query for all of the **shots** of Movie A which are annotated by compound or glommed icons containing components of type X. When a particular Movie is filtered in this manner, the results are Media Time Line Icons which represent the shots of the Movie which satisfy the query. The query shown would return, in the form of Media Time Line Icons, all of the **shots** of Movie A which contain annotations of a "land vehicle."

After icons have been retrieved, they can be sorted in the Icon Palette using the following Sort Buttons:



alphabetically



by type (i.e., position within the Icon Workshop hierarchy, and by the distinction between Glommed versus Compound icons)



by annotator



by frequency of use



by creation date



by length (applicable only to Time Line Icons)



by density (applicable only to the logs represented by Time Line Icons)



by chronological order (applicable only to Time Line Icons representing sequences within a single movie)

Once an annotator has made and/or found the descriptors needed to annotate, the next step is to use these atemporal descriptors to make temporal descriptions of video content on Media Time Lines.

6.2.5. Making Descriptions in Media Streams

The Media Time Line is the interface component used in Media Streams for annotating video content. The Media Time Line separates annotation into various streams. These annotation streams reproduce and refine the icon categories of the Icon Space. They enable the icons used in annotation to be *viewed in context*. This design principle, like the hierarchical organization and compounding of the Icon Space, enables the large number of Media Streams' compound and glommed icons to maintain their intelligibility.

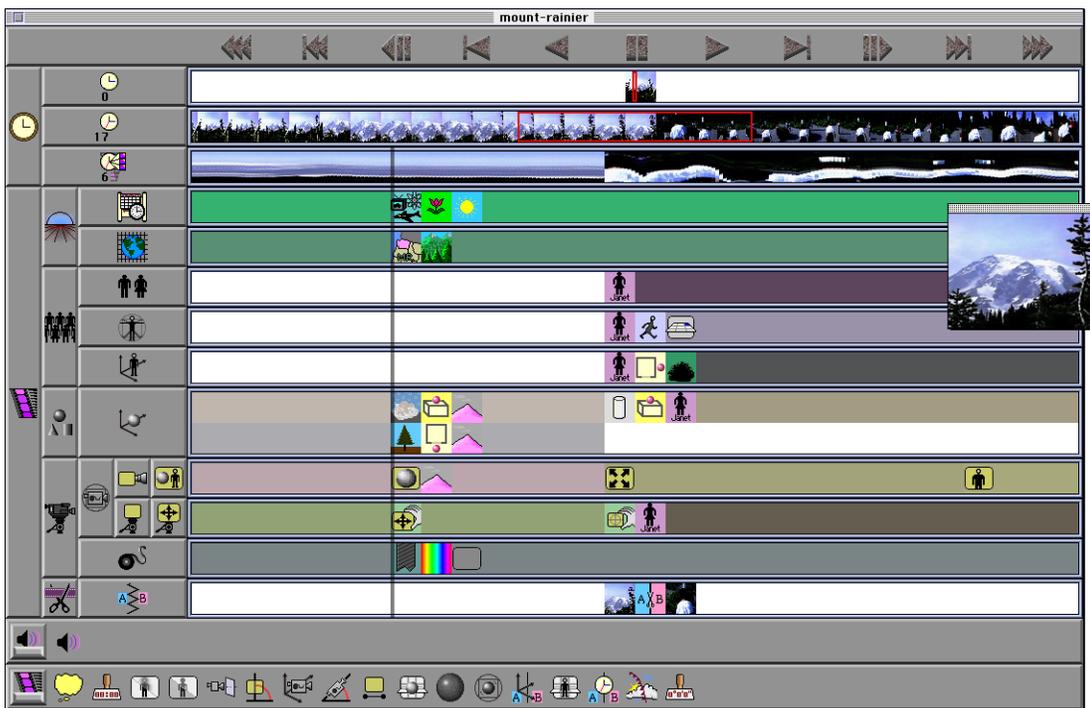


Figure 72. The Media Time Line

Annotations are made by dropping icons onto a Media Time Line. The Select Bar indicates the current frame position and the icons currently valid at that frame. This is especially useful in the common case in which the beginning of an annotation is no longer visible:

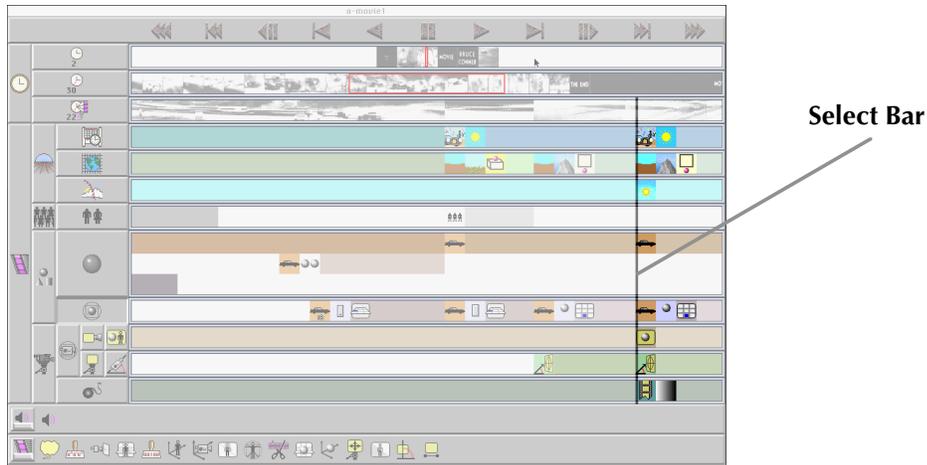


Figure 73. The Select Bar (insertion point) on the Media Time Line

In most cases, icons know which annotation stream to go in because of their type. When dropped on the Media Time Line, icons either will begin an annotation whose extent is readable by the color bar after the icon, or be glommed onto an existing and available annotation in order to form a glommed icon (glommed icons are icon sentences made out of compounds, e.g., “An adult female doctor holds a stethoscope”). Annotations are “good-till-canceled” meaning that they extend either until the end of a shot or until the end of the Media Time Line (this is adjustable on the Settings Control Palette). To set the end point of an annotation, one can either drag its representation off the Select Bar or use the standard Macintosh cut commands. The start and end points of annotations are also adjustable.

The various annotation streams of the Media Time Line provide the context and intelligibility for our stream-based representation of video content. There are, however, many of them, so the Media Time Line supports the hiding and unhiding of annotation streams in special hide bars at the bottom of the Media Time Line.

The annotation streams are structured hierarchically to provide multiple contextual levels. These hierarchical annotation streams can be expanded and collapsed, hidden and unhidden, and rearranged within their level. The entire structure of annotation streams is as follows:

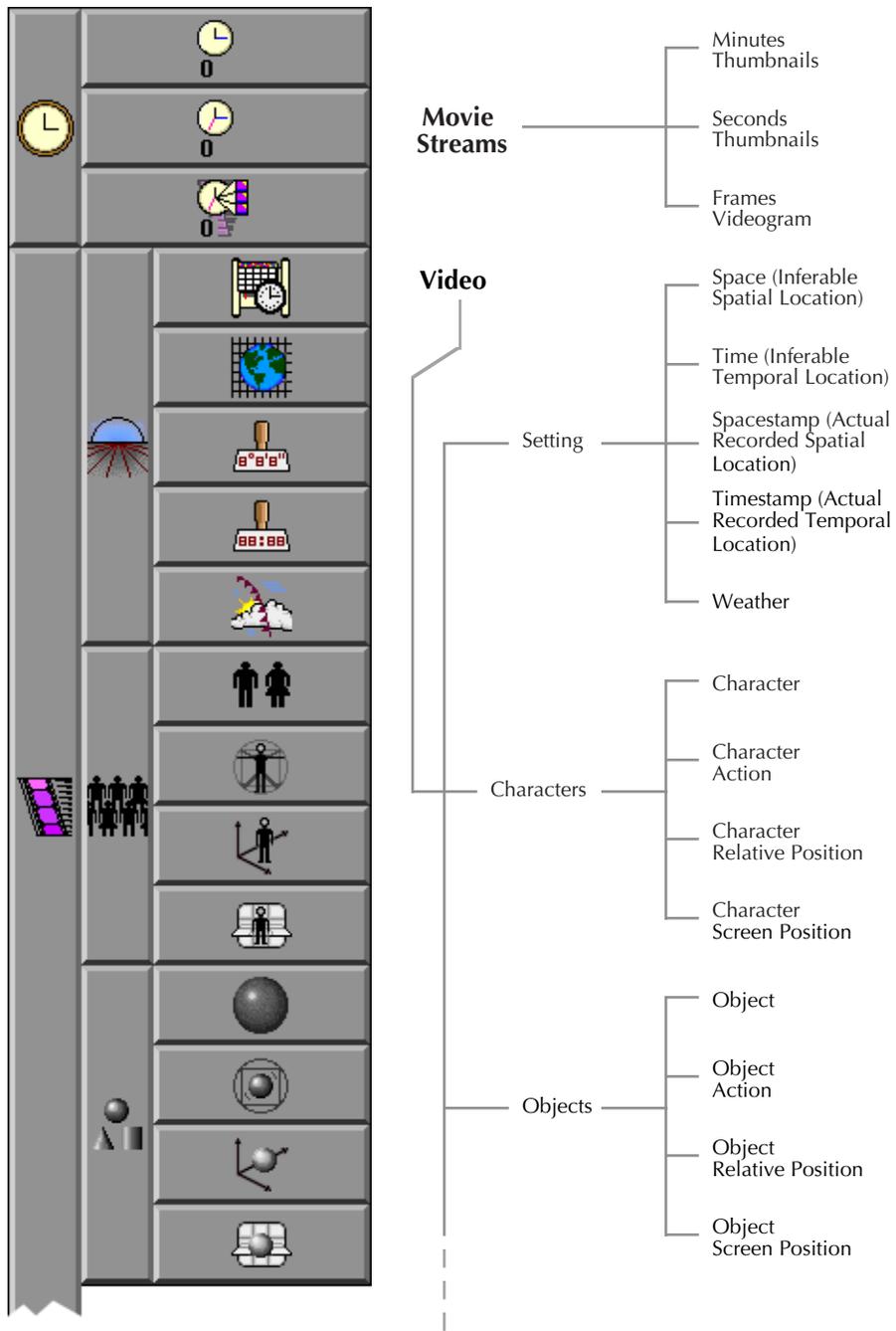


Figure 74 A.

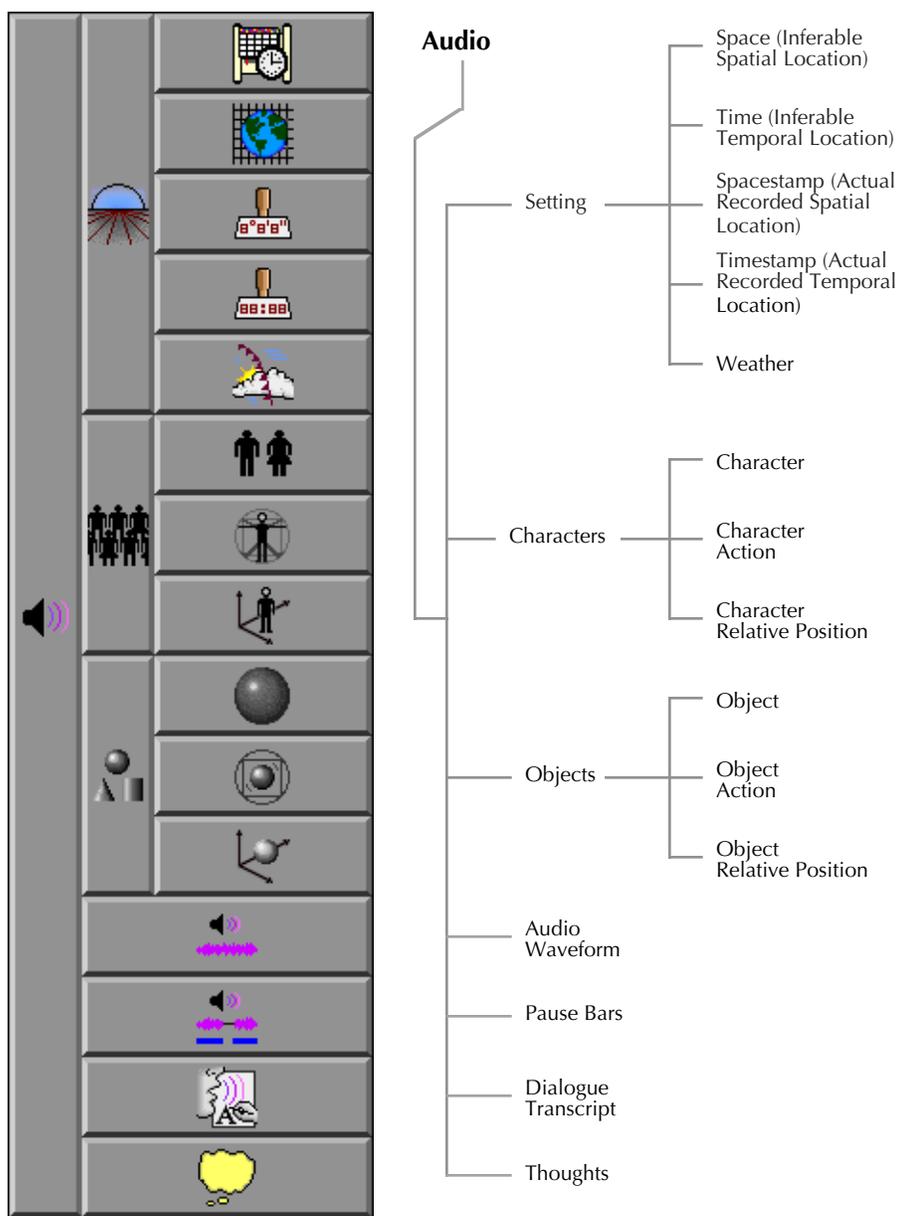


Figure 74 C.

We have noticed that annotators have different styles of working with the Media Time Line. The design of the hierarchical annotation streams supports two very different approaches to annotating a video: vertical logging (move through the video by logging all annotations streams for a shot and then moving on) and horizontal logging (log one or a group of annotation streams for the entire video and then start again on another group). The later strategy especially supports division of labor in the annotation process by enabling different people to focus on different aspects of the content.

In Media Streams, audio and video are annotated separately. This solution is a compromise. Ideally, one would annotate a movie three times: once for the audio only, once for the video only, and once for the meaning of the combined audio and video streams. In order to avoid that very labor-intensive task, we annotate the audio and video separately so as to enable them to be repurposed independently. The audio representation in Media Streams is very minimal since our focus has been predominately on the semantics and syntax of the moving image. What we have done is offer two visualizations for audio (waveform and pause bars) as well as the ability to easily transcribe dialogue. We also enable the annotation of audio events using our iconic descriptors.

In annotating the presence or absence of audio events within the data stream, our representation makes use of the fact that in thinking about audio, one thinks about the source that produced the audio. In order to annotate audio events, one uses icons for different objects and characters which are glommed with the icon for the action of producing the heard sound. This concept correlates to Christian Metz's notion of "aural objects" (Metz 1980). Many other representations of audio are possible and necessary. Bordwell outlines several (Bordwell and Thompson 1990: Chapter 8). An important distinction that we implicitly support is the difference between on-screen and off-screen sound. We annotate the sound source in the audio and the sound source in the video *separately*. If a sound source appears in both streams it is on-screen (sound producing actions annotated in video are automatically reproduced in audio), if a sound source appears only in audio, it is off-screen.

In Media Streams, it is annotation that makes content-based retrieval and repurposing of video possible. Annotations on the Media Time Line also enable new forms of content-based browsing.

6.3. Browsing

Retrieval is the search for content one does not presently have; browsing, however, is the often non-goal-directed search through content one does have. One can browse to get an overview of a piece of media or to find specific content in the media. We can identify three modes of browsing temporal media that are especially useful:

- **FAST FORWARD**

All frames are played but at a faster than real-time rate. Most fast forward systems simply speed up the frame rate. There has been some work on adjustable fast forward that takes content into consideration in setting a variable playback speed (Akutsu and others 1992). Advanced fast forward systems would be like a car on a road. The feel of the road gives the driver of the car information about the terrain and speed can be adjusted according to this information. A video fast forward system would adjust its speed according to the content of the video, slowing down and speeding up appropriately.

- **SKIMMING**

Only selected frames are played and at variable rates. Like flipping pages in a book or speed reading, skimming video is the selected presentation of parts of the video stream as one moves through it at variable rates, though the total skimming time is usually faster than real time.

- **JUMPING**

Advances to a specific segment of frames in the media stream and plays that segment of frames. Like the track advance button on a compact disc player, jumping through video would advance from the current position to the next logical unit and enable the playing of that logical unit (e.g., a shot, a segment at a certain spatial location, etc.).

Media Streams supports all three modes of browsing. We have all of the standard videotape deck control buttons and a few non standard ones which will be discussed below. Media Streams supports standard fast forward through scan forward and scan reverse controls.



Figure 75. The eleven Movie Controls. From left to right, they are: Scan Reverse, Jump Reverse, Frame Reverse, Reverse Play Extent, Reverse Play, Pause, Play, Play Extent, Frame Forward, Jump Forward, and Scan Forward.

We support skimming in two different ways: through “scrubbing” (tightly coupled linkage between a moveable interface object and frame position) and through multiple timescale representations. Media Streams enables users to scrub through video at three different rates. The Select Bar supports *frame* level scrubbing. The *seconds* and *minutes* thumbnail movie streams each has their own scrubber that enables users to skim through content at different sampling rates.



Figure 76. The Minutes and Seconds Scrubbers

On their own these simultaneously displayed, multiple timescale movie thumbnails and videograms also support skimming by enabling users to get an overview of video content and surrounding context at any point in time. Thumbnails also support jumping. A user can double-click on any thumbnail and jump to that point in the video. Media Streams also supports content jumping through two new button controls which enable users to jump to and play any video segment according to its annotated content or according to its automatically extracted segmentation. We add jump forward/reverse and play extent forward/reverse controls to the standard palette of video controls. The Movie Controls of the Media Time Line are explained in the following table:



Jump Forward..... (Jumping by content) Jumps the current frame to the next change in the selected stream(s). If no stream has been selected, this control will advance the current frame to the next shot boundary.



Jump Reverse..... Jumps the current frame to the previous change in the selected stream(s). If no stream has been selected, this control will jump the current frame to the prior shot boundary.



Play Extent..... Plays that segment of the movie whose start-frame and end-frame are determined by an annotation or pair of shot boundaries, and which contains the current frame. If no stream is selected, this control will play the current shot (defined by the shot boundaries immediately bracketing the current frame — often discernible in the Videogram, and represented elsewhere in the Transitions stream). If a stream has been selected, this control will play the segment of the Movie that corresponds to the annotation in the selected stream and contains the current frame. If there is no annotation in the selected stream at the current frame, nothing will be played. If the current frame has multiple annotations that describe it (as could happen if multiple streams were selected or if the selected stream was expandable), the "Play Extent" control will play from the beginning of the first current annotation to the next change in the selected stream(s).



Reverse Play Extent..... Reverse-plays the segment of the movie containing the current frame and whose endpoints are defined by an annotation in a selected stream or the shot boundaries bracketing the current frame.

These buttons would allow (digital or digitally buffered) VCRs to have the ability to jump to the next shot and/or play the next shot. With annotated video, it becomes possible to jump to the next content segmentation—the next new character, change in location, camera move, etc. The play extent buttons also enable users to play only those parts of a video in which a certain annotation occurs. They provide very granular and content-based

control over playback. In many ways, the Media Time Line with its various interface objects and controls is like a *content VCR*.

The Media Time Line is Media Streams' main interface object for annotation and browsing. It is also where retrieval queries for video are formulated. The challenges in designing a query interface for temporal media are manifold and complex. We address these issues in the next section.

6.4. Retrieval and Repurposing

The purpose of video representation and annotation is to enable the retrieval and resequencing of video content. Interfaces for video retrieval are in the very early stages of development. In this section we map out the interface issues video retrieval systems will have to address.

Retrieval can be thought of as the process whereby a user or a program initiates an action that results in a desired piece of video being selected from (or made out of) existing video in a video database. Since video is opaque and requires representation, the action that initiates the selection of the piece of video from the database is itself an act of representation. This act of representing the video to be retrieved is known as *query formulation*.

The process of query formulation is deeply related to the process of annotation. Annotation is describing video one has; query formulation is describing video one wants to have. The same interface may be used for both. However, query formulation is not identical to annotation. In annotation, the relationship between the description and the object of description is, at the time of annotation, one-to-one. In query formulation, the relationship between the description and the object of description is at least potentially one-to-many. Many possible video segments may match a given query because a query by its very nature is a description that may find a match that is *similar* to the query, not only the one that may be *identical* to it. Therefore, an interface for query formulation needs to enable a user to control the parameters which determine the ways in which the similarity of the query and its possible matches is formulated.

There are two primary types of query formulation: *query by example* and *query by description*. Media Streams supports both forms of query for annotated video. It is query by description that is the central concern of query formulation in Media Streams because of the similarity between annotation and query formulation on the Media Time Line.

6.4.1. Query By Description

Query by description is the most common form of query used today. A user formulates a description of the content to be retrieved in some query language. This language may be SQL, a natural language or a restricted subset of one, a series of keywords, or as in Media Streams, an iconic visual language that supports temporal and other relationships between the descriptors. Since video must be described in order to be retrieved according to high level features of its content, the query language can function as a superset of the annotation language.

6.4.2. Query By Example

Another very powerful form of query that does not have to rely on the specific formulation of a detailed query is *query by example*. If all videos in a video database have an attached description of their contents—bits about their bits—any video segment or sequence can itself be used as a query. For users untutored in the query language, this query by example mechanism can be a powerful and intuitive form of query.

6.4.3. Image Retrieval vs. Video Retrieval

Most query languages and retrieval interfaces used for video today were derived from, or simply are, query languages and retrieval interfaces developed for other media (most often text or still images). However, video is a temporal medium with a unique semantics and syntax. Therefore, any query language and retrieval interface used for video must grapple with some fundamental interface issues which arise from the specific properties of the medium. The most obvious issues which need to be confronted are those which are due to the *temporal* structure of video data.

6.4.4. Issues in Temporal Query

In designing an interface for video retrieval the visualization and manipulation of the temporal structure of video data is a primary challenge. Let us begin with an abstract example of video query to elaborate the issues involved. Imagine a video query that extends from time t to time $t+n$ and contains the annotations x , y , and z :

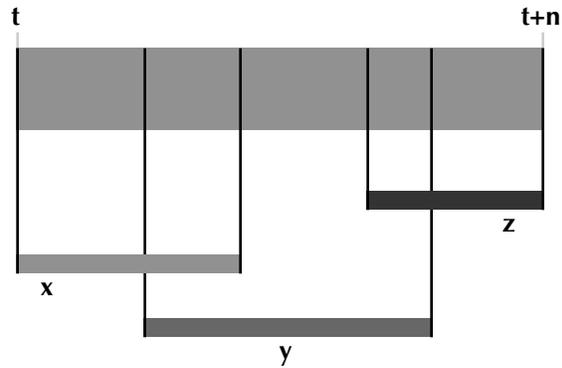


Figure 77. A query of three annotations

This query may seem straightforward and simple enough, yet it is rife with questions which an interface for video retrieval must answer:

- what *temporal duration* is being described in the query (screen duration, plot duration, story duration)?
- what *temporal scale* does the query represent (30 frames per second, 1 frame per minute, etc.)?
- what *temporal relations* are expressed by the ordering and length of annotations **x**, **y**, and **z** in the query?
- what *continuity relations* are implied among **x**, **y**, and **z** in the query? In other words, what if the retrieved matches for **x**, **y**, and **z** come from different shots or from different movies?
- what *relative importance* is respectively assigned to **x**, **y**, and **z** in the query?
- what *similarity criteria* are used for the matches to **x**, **y**, and **z** in the query?

In order to begin to deal with the challenge of interfaces for video query let us examine each of these issues in more detail.

6.4.4.1. Temporal Duration

The first question one must ask in a temporal query system for video data is “What kind of time is the query describing?” The answer here relates to our earlier discussion of the three types of temporal duration described by David Bordwell (Bordwell 1985: 60). A temporal query into a video database could be querying about temporal durations taken from any of the three temporal orders: screen time, plot time, and story time. In Media Streams, all temporal durations refer to screen time—the actual duration of events on the video screen. One could imagine temporal query systems which enabled users to access video according to the two other temporal orders. That would be possible if information about the event structure and event durations were encoded for story and plot time. A system with such encodings would have three different notions of the temporal duration of a query.

The interface for query formulation also needs to deal with the durations of the query parts themselves. In most cases, users are expressing temporal relations between query parts and approximate rather than exact durations. But this is not always the case. In highly synchronized music videos, for example, absolute temporal precision may be of the utmost importance. Having a way to specify the “fudge factor” of the durations of query parts is an open issue for temporal query interfaces.

6.4.4.2. Temporal Scale

For temporal scale to be meaningful in a video query there are two simple and rather obvious guidelines:

- the descriptors in the query which appear to exist in the same time scale should actually do so
- the time scale needs to be indicated clearly to the user

If the abstract query example above is interpreted with respect to 30 frames per second it will retrieve drastically different video segments than if it is interpreted with respect to 30 frames per hour.

A more complex and subtle issue in representing the time scale of the query is the meaning of the indication of temporal duration within a given time scale. As an illustration, let us take our abstract example and add a definite time scale to it: 30 frames per second. Let us make the overall duration of the query 100 frames ($t = 0$, $t+n = 100$). The respective frame lengths of annotations x , y , and z would be 39, 44, and 40 frames. At the time scale of 30 frames per second the difference among 1.298, 1.474, and

1.333 seconds may not seem like much. (There are notable exceptions of course in which frame accurate lengths are often essential. In carefully edited and timed video sequences such as music videos differences of even one thirtieth of a second are significant.) But imagine if the time scale were changed to 1 frame per minute. Then the respective durations of annotations **x**, **y**, and **z** would be 39, 44, and 40 minutes. The question then arises whether the necessary visual precision of the pixel-based interface affords the accurate communication of the appropriate accuracy of annotation durations at various time scales. In other words, sometimes users want to formulate frame accurate durations in video queries, and sometimes they want to convey durations which avoid overly restrictive temporal precision. Determining how to give the user access to this parameter and deciding when to have the machine handle the precision of temporal durations as a function of the time scale and the matches in the video database is a problem which current video retrieval systems, including Media Streams, have yet to address.

6.4.4.3. Temporal Relations

The problem of an interface for video query that supports retrieval on temporal relations between query parts has an affinity with the above mentioned issue of the temporal precision of the parts of a video query. Quite simply, there is a significant difference between queries which represent a certain set of imprecise but coherently ordered temporal relations between query parts and queries which represent the precise numeric temporal extents of query parts and their respective temporal relationships.

If we return to our abstract query example above, using a time scale of 30 frames per second, the temporal relations of the query parts can be interpreted as precise numerical relationships or as approximate symbolic ones. If the query is interpreted as conveying a strict numeric order and duration of the annotations **x**, **y**, and **z**, then the query could be expressed as follows:

Find me a video stream of 3.33333 seconds in length in which **x** starts at 0.00000 seconds and ends at 1.29824 seconds, **y** starts at 0.77192 seconds and ends at 2.24561 seconds, and **z** starts at 2.00000 seconds and ends at 3.33333 seconds. (This numeric precision may seem fanciful, but in many audio editing applications time units measured in milliseconds are not uncommon.)

If, however, the query is interpreted as conveying a set of approximate temporal relations among the query parts, then the query could be expressed as follows:

Find me a very short (about 3 seconds in length) video stream in which **x** starts at the beginning and ends after **y** starts and before **y** ends, **y** ends after **z** starts and before **z** ends, **x** ends before **z** starts, and **z** ends at the end.

The numeric representation can be converted into one exact set of symbolic relationships—the reverse is not true. The set of symbolic relationships, however, can be expressed by a potentially infinite set of numeric ones (though not useably infinite since the granularity of temporal distinctions that matters to human perception is not infinitely small). However, without a numeric grounding the situation becomes far more complex. If the symbolic relationships are not expressed with completeness (i.e., all of the relationships between the temporal extents are either enumerated or inferable from the symbolic expression), they can even result in potential ambiguities. Visual temporal query interfaces have the advantage of compelling the user to make the temporal relationships explicit, the danger, however, is that the precision, completeness, and restrictiveness of these temporal relationships may be the result of false implicature. Media Streams addresses the computational issues for temporal relationships by using both numerical and symbolic representations, but the user interface problems surrounding their formulation are far from solved.

6.4.4.4. Continuity Relations

In the above example, part of the query is asking for an **x** that overlaps a **y**. This **x** could be satisfied by two **x**'s. Imagine a video sequence in which the first shot has an **x** without a **y**, and the second shot has an **x** that starts a **y**. This sequence of shots would satisfy the temporal relation of an **x** overlapping a **y**. But that **x** would be a sequence of **x** from Shot 1 and **x** from Shot 2. Is that a permissible answer to the query?

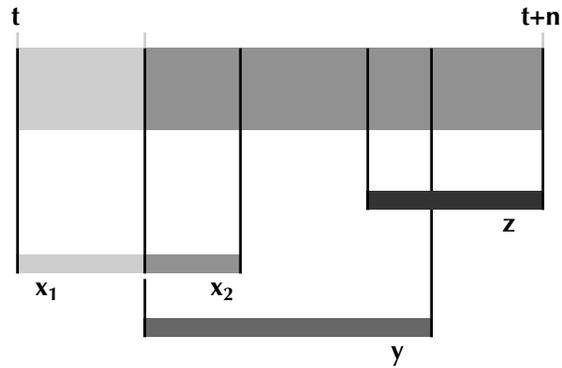


Figure 78. Issue of Identity in Video Sequence Retrieval

The issue here is a deep one. What forms of continuity are or are not expressed in a temporal query? What synthetic results are allowed to satisfy a temporal query into a video database? What constitutes continuity of descriptors across shots and across movies for a descriptor in the database? Media Streams' representation attempts to answer these questions in the design of its representational categories, memory structures, and retrieval algorithms. In its query interface it assumes that retrieval-by-composition methods are allowed. An important and needed extension to the interface is the ability to distinguish between query parts in which the *identity* of descriptors is being called for versus the *similarity* of descriptors.

6.4.4.5. Relative Importance of Query Parts

Interfaces for video query also need a way of expressing the relative importance of query parts so that the user has control over the output of the retrieval algorithm. For example, in a query for video segments in which the query contains two parts: close up and adult male, the relative importance assigned to matches retrieved for each part would result in two different sets of matches. In the case in which the close up part of the query is more important than the adult male part of the query, matches would receive higher scores for satisfying that part of the query than the other. In the case in which a close up is more important than adult male: a close up of an adult male would be the best match, a close up of a middle-aged male would be a less good match and a medium shot of an adult male would be an even less good match. In the case in which adult male is given greater relative importance in the query, the results would be different: a close up of an adult male would still be the best match, a medium shot of an adult male would be a less good match, and a close up of a middle-aged male would be an even less good match. In Media Streams, the ability to arrange annotation streams affords the ability for

users to set relative priorities for query parts. In order to be able to use this feature of the interface, we will have to revise our scoring mechanism so that semantic scores can be a single number that can be scaled by the relative importance of the query parts.

6.4.4.6. Similarity Criteria

A final issue for temporal query interfaces is the user control over the various types of similarity that can be expressed in a query: semantic, relational, and temporal. The problem of treating each of these similarity dimensions as parameterizable constraints on retrieval and being able to control them through the query interface is an open and intriguing research issue in Media Streams.

What makes the interface problem for video query so difficult is that all of the above factors may influence each other. Designing interfaces in which multiple mutually constrained video query dimensions are accessible to user control will demand rigorous research of which our work is only a first step.

Media Streams' main contribution to interfaces for video query is that the interface for annotation and query are the same. This conjunction enables the tasks of annotation, retrieval, and repurposing to interleave in ways not possible before.

6.4.5. Media Streams Retrieval Interfaces

Video queries are formulated on Media Time Lines (and to a more limited extent within the Icon Query Language on the Query Bar). The interfaces for viewing the results of these queries have already appeared in Chapter 5. In response to user queries, Media Time Line Icons representing the retrieved video segments and sequences appear in the Result Region of the Icon Palette. Media Time Line Icons are playable movies and with a control-click reveal the explanation of their matching criteria in Score Windows.

6.4.6. Query Refinement

Media Streams supports query refinement through the parameterization of three aspects of retrieval:

- **Semantic Threshold**
(the number of matches per compound that should be considered)
- **Completeness Threshold**
(how complete a partial match has to be to be valid)
- **Matches Threshold**
(the number of matches to be shown as Media Time Line Icons in the Result Region of the Icon Palette)

The interface for controlling these parameters uses simple sliders and numerical representations:



Figure 79. The Search Control Palette

The design of interfaces which provide control over other parameters affecting query and retrieval (many of which were mentioned in previous sections) is ongoing.

6.4.7. Learning from Retrieval: The Analogy Editor

In the last chapter we discussed Media Streams' ability to index episodic exceptions to its semantic ontology. The interface for teaching the system these exceptions is the Analogy Editor. Let's work through our Charlie Chaplin example to see how we tell Media Streams that sometimes (as when Charlie Chaplin eats one) a *shoe* can function as *food*.

We formulate the query for “a character of indeterminate sex eating a food object.”



Figure 80. A simple query

We retrieve the following video segments:

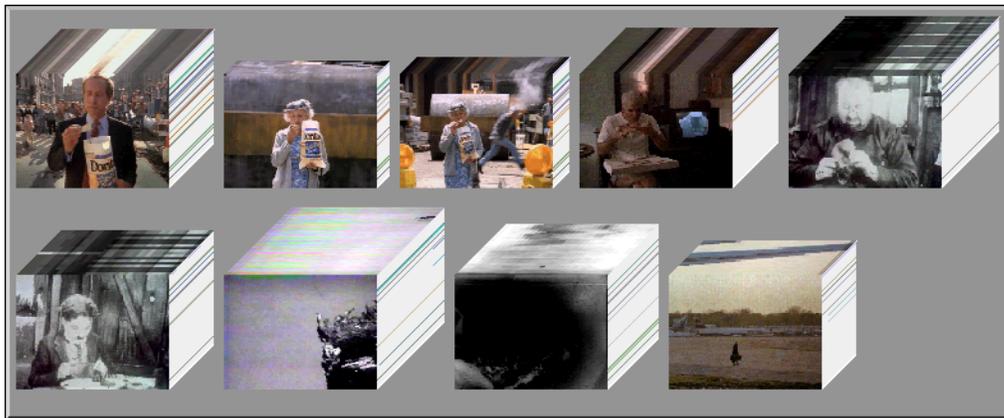


Figure 81. The results from the query in figure 80

We see “Chevy Chase eating Doritos,” “An elderly female eating Doritos,” another video segment of the same elderly female eating Doritos, “Steve Martin eating pizza,” “an adult male eating a shoe,” “Charlie Chaplin eating a shoe,” and then three video segments of characters of indeterminate sex doing different things.

In order to tell the system that “Charlie Chaplin eating a shoe” is a good match to “a character of indeterminate sex eating a food object,” we call up the Analogy Editor by placing the Media Time Line Icon for “Charlie Chaplin eating a shoe” inside a waiting part of the Query Bar. The Analogy Editor pops up telling us what the current prototype relations are between the terms of the query and the terms of the match.

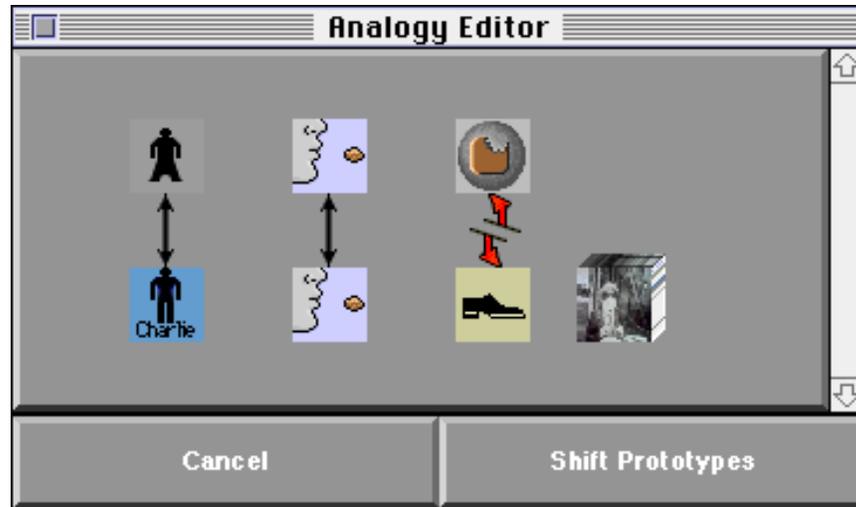


Figure 82. The Analogy Editor

As we can see in the Analogy Editor above, *Charlie Chaplin* and a *character of indeterminate sex* have a common prototype, *eating* and *eating* have a common prototype, but *shoe* and *food object* do not. By clicking on the relation between *shoe* and *food object* we can tell Media Streams that this particular instance of *shoe* (here functioning as a *food object*) should be indexed under the Indices of *food object* in the CIDIS.

The Analogy Editor changes the appearance of the relation between *shoe* and *food object* to reflect what will happen when the prototype of *shoe* is shifted to *food object*.



Figure 83.

After shifting the prototype of *shoe* and reindexing, we will get a different ordering of the results if we perform the query again:

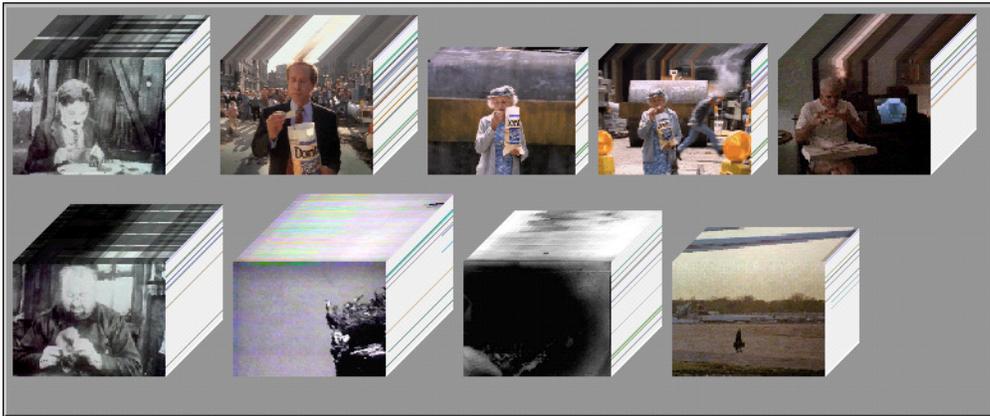


Figure 84. The results of the query in Figure 80 after shifting the prototype of *shoe*

One may wonder why the (other) video segment of “an adult male eating a shoe” (in the lower left corner) was not also affected by the shifting and reindexing of the shoe that Charlie Chaplin was eating. We have to remember the nature of the claim we are making about learning from retrieval: Learning from retrieval means being able to index human-tutored examples of sequence-dependent exceptions to the semantics of the ontology and being able to use these indexed exceptions in later search and retrieval. In order to go beyond this functionality we would have to solve the problem of how to propagate the new indexing knowledge throughout the semantic memory such that existing episodic examples of *shoe eating* and new examples of *shoe eating* would be interpreted in light of the reindexed example of the *shoe Charlie Chaplin was eating*. This problem is reminiscent of the developmental processes in human cognition of *assimilation* and *accommodation* (Gruber and J.J. 1986). Developing computer programs which can exhibit the ability to reinterpret past examples in terms of new ones (accommodation) and/or reinterpret new examples in terms of past ones (assimilation) will require the development of episodic memory systems which are more dynamic than the scheme Media Streams uses of indexing episodic examples within our semantic memory. What is required is the ability to analogize from examples to examples so as to remap the relations and the semantics within them. This is a very hard problem which Mnemosyne itself is grappling with: how do we know what to map from one example to another, i.e., how do we generalize from examples? How far and in what ways can new experiences reconfigure existing knowledge? This process is what Seymour Papert considers the engine of cognitive development:

Papert's Principle: *Some of the most crucial steps in mental growth are based not simply on acquiring new skills, but on acquiring new administrative ways to use what one already knows. (Minsky 1987: 102).*

This problem is certainly something to chew on.



In this chapter we have outlined several of the innovations of Media Streams' user interface for visualizing, annotating, browsing, retrieving, and repurposing video. What separates Media Streams most obviously from other approaches to video annotation and retrieval is the development and use of our iconic visual language. In the next chapter, we discuss the arguments for and against using an iconic visual language for video annotation and retrieval.



Chapter Seven

Why Icons?

7. Why Icons?



The most obvious and unique feature of Media Streams' user interface is its iconic visual language for video annotation and retrieval (Davis 1993a; Davis 1993b). The representation and retrieval structures in Media Streams could be manipulated by many types of human-computer interface; however, the choice of an iconic visual language for this task is not an arbitrary or unimportant one. It represents a solution for the design of practical video annotation systems today as well as a statement about the future of systems for media manipulation. By decreasing the tedium and increasing the reusability of annotation effort, Media Streams' iconic visual language may solve many of the current problems of the stock footage industry whose antiquated technology and practices are inadequate to the task of on-time and accurate retrieval of video data (Greenway and Mouchawar 1994). A uniform and widespread iconic visual language for video annotation and retrieval will enable the creation of a global media archive in which video can be stored and reused.

Media Streams' iconic visual language also points toward the development of new forms of visual literacy which will become predominant in the coming age of computational media. We are currently in a crucial phase of a second "Gutenberg shift" (McLuhan 1962) in which video is becoming a ubiquitous data type not only for viewing (i.e., reading) but for daily communication and composition (i.e., writing). This shift will only be possible when we can construct representations of video which enable us to parse, index, browse, search, retrieve, manipulate, and sequence video according to representations of its content. These representations of visual media will themselves be visual. An iconic visual language for video annotation and retrieval will support new forms of video writing (repurposing of video content) within a widespread practice of asynchronous many-to-many daily video communication.

7.1. Iconic Visual Languages

There have been serious efforts to create iconic languages to facilitate global communication (Bliss 1978; Neurath 1981) and provide international standard symbols for specific domains (Dreyfuss 1972). We developed Media Streams' iconic visual language in response to the challenge of creating a language for representing video content for retrieval and repurposing. Our iconic visual language is computationally writable and readable, and makes use of a structured, controlled, searchable, generative vocabulary of iconic primitives.

- It is *structured* due to the semantic hierarchical organization of primitives in the Icon Workshop.
- It is *controlled* because it has a limited vocabulary in which new expressions are created in terms of existing ones.
- It is *searchable* due to the tripartite structure of CIDIS, Media Time Lines, and Indices.
- It is *generative* due to the compositional semantics of compounding in the Icon Workshop and glomming on Media Time Lines.

It is designed to support the task of representing the semantics and syntax of video data for retrieval and repurposing by diverse communities of users. Current video annotation and retrieval systems use textual representations which are inadequate to this task. The reasons that Media Streams uses an iconic visual language rather than keywords or natural language are discussed in the next section.

7.1.1. Keywords vs. Iconic Visual Language

As described in Chapter Two, video archives which use keyword-based annotation cannot represent the *semantic*, *relational*, and *temporal* content of video data. Moreover, keyword-based systems cannot achieve *convergence* of descriptions by different annotators or *scale up* to very large archives with great variety of content. Current archival practice does not have the tools which would enable it to overcome these limitations. To remind us of some of the difference between keyword descriptors and Media Streams' iconic descriptors we can examine the simple example in which we want to describe a video of "a dog biting Steve." A keyword-based description would have the following information:

**dog,
biting,
Steve**

These keywords are not *semantic* terms. We do not know that "a dog is a mammal" or that "a Chow is a dog." We do not know the *relationships* between the descriptors. Is this description of a video in which "a dog

bites Steve” or “Steve bites a dog” or in which “Steve is biting and a dog is in the scene” or in which “a dog is biting and Steve is in the scene”?

In Media Streams’ iconic visual language, the video of “Steve biting a dog” would have the following iconic descriptors attached to it:

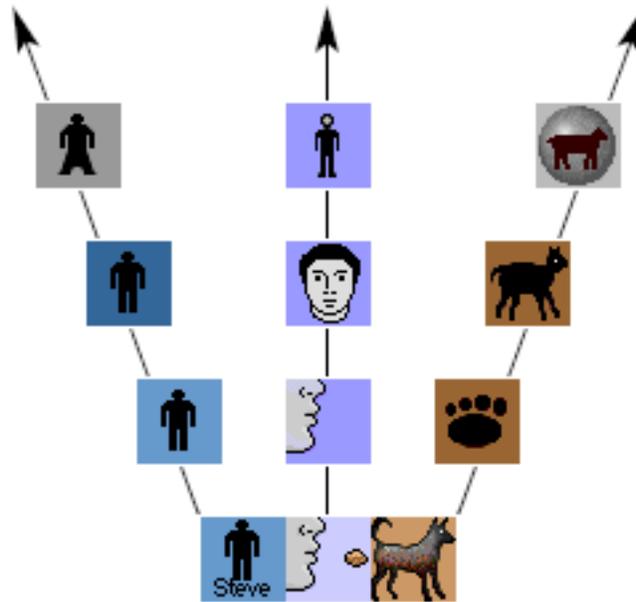


Figure 85.

The bottom glommed icon is the iconic descriptor for “Steve biting a dog” but it is not alone. Its component terms are embedded in a semantic hierarchy that defines their similarity to other terms. We know that Steve is an adult male, that biting is a mouth action, that a dog is a pawed mammal, etc. The relationship between Steve, the dog, and biting is also clearly expressed by the syntax of the glommed icon. Furthermore, iconic descriptors, unlike keywords, have a *designed similarity of terms*. The terms that are *semantically* similar to Steve, biting, and dog are represented by icons which are *visually* similar to them through the use of color, shape, and animation.

Keywords are clearly inadequate to the task of functioning as a structured, controlled, searchable, generative vocabulary for video representation. One may argue, however, that keywords are an overly restricted example of a textual representation and that it is natural language that is better suited to video representation than an iconic visual language.

7.1.2. Natural Language vs. Iconic Visual Language

Current computer systems cannot process natural language sufficiently well to use it as a semantic, relational, temporal representation for video content. Syntax parsing has made some headway (Haase 1991), while semantics parsing is improving for small passages of text but breaks down for long, complicated passages (Chakravarthy 1995; Lenat 1994a). However, the current level of technology is not the standard by which to judge the inadequacy of natural language. Even if we assume that we have sufficiently robust computational parsing of natural language, it is still inferior to an iconic visual language for the task of video representation. Let us consider the following example. Imagine in your mind's eye the video described by the following passage:

Jack, an adult male police officer, while walking to the left, starts waving with his right arm, and then has a puzzled look on his face as he turns his head to the right; he then drops his facial expression and stops turning his head, immediately looks up, and then stops looking up after he stops waving but before he stops walking.

Got it? How long will it take you to see the temporal and spatial relationships of the body movements described in the above passage? Even if you succeed in visualizing these relationships are you certain about their precise relations?

Now look at the portion of the Media Streams Media Time Line below:

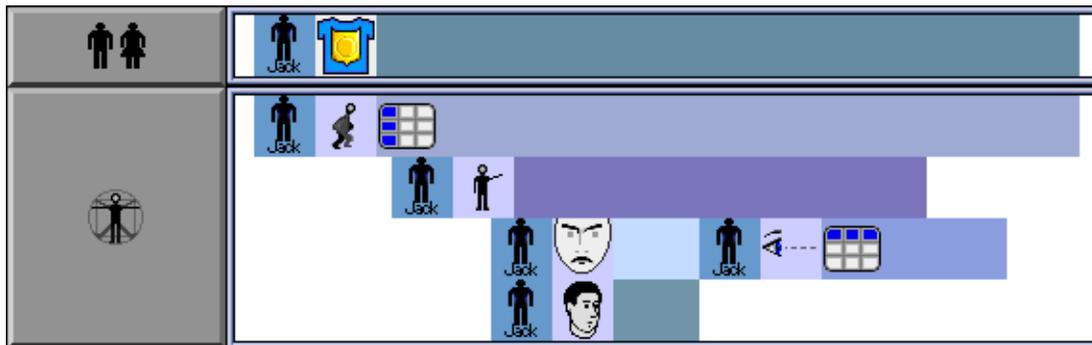


Figure 86.

In Media Streams' iconic visual language the temporal and spatial relationships of body movements are clearly represented. Natural language is deficient as a medium for describing complex temporal sequences of actions, expressions, and spatial relationships. One may argue that even natural language is a straw example of a textual

representation and that a designed, stream-based textual representation would be superior to an iconic visual language.

7.1.3. Text vs. Iconic Visual Language

There are two possible paths to creating a designed textual representation for video: a structured subset of an existing natural language or the invention of an artificial textual representation. The first case would use existing words from natural language as signs within a designed and structured representational system. Many knowledge representation languages take this form. The above log could be represented as:

```
{[Adult Male (Jack) - Police Officer]-----}
{[Adult Male (Jack)] : [walks] : [screen left]-----}
  {[Adult Male (Jack)] : [waves left arm]-----}
    {[Adult Male (Jack)] : [puzzled look]}{[Adult Male (Jack)] : [looks] : [screen up]----}
      {[Adult Male (Jack)] : [turns head
        to the right]}
```

Figure 87.

The problem with using a subset of natural language for this task is that terms which should be seen to be similar are not necessarily similar. For example, *turns head to the right* and *puzzled look* are both head actions, but that commonality is not visible in these phrases taken from natural language. We must then turn to the second and most theoretically extreme case of a textual representation: a designed and structured textual representation system for video. In such a system, textual terms would be designed to capture salient features and similarities in the vocabulary as is the case in iconic visual languages.

As an experiment, let's examine two compound icons from the above log (*[Adult Male (Jack)] : [puzzled look]* and *[Adult Male (Jack)] : [turns head to the right]*) and create a designed and structured textual representation, let's call it *LingVid*, for them:

LingVid:

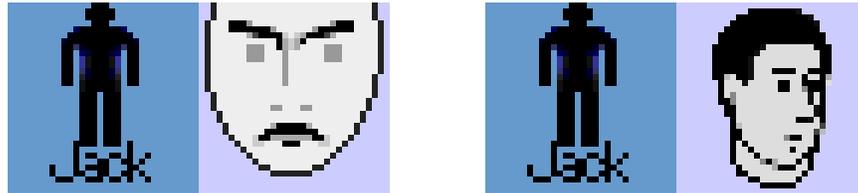
Jackvam-parosgauwon

Jack	<i>Jack</i>
va	<i>adult</i>
m	<i>male suffix</i>
pa	<i>face</i>
ros	<i>head</i>
ga	<i>eyebrows</i>
u	<i>distal endpoints up</i>
wo	<i>mouth</i>
n	<i>distal endpoints down</i>

Jackvam-rosvirdax

Jack	<i>Jack</i>
va	<i>adult</i>
m	<i>male suffix</i>
ros	<i>head</i>
vir	<i>turn</i>
dax	<i>right</i>

Media Streams:



LingVid's textual representation exhibits the designed and structured semantic, relational, and temporal features of Media Streams' iconic visual language. However, there still exist significant and decisive differences between LingVid and Media Streams which argue for the use of an iconic visual language for video annotation and retrieval. These differences relate to the distinct representational affordances of visual as opposed to textual systems. The first distinction is the serial versus parallel legibility of text versus images. The predominantly serial organization of textual systems has difficulty supporting the forms of parallel legibility which images enable:

- Gestalt View of Features
- Foregrounding and Backgrounding
- Spatial Relationships

The parallel legibility of images is also a property of the medium they are used to represent: video. This homology points to the strongest and more subtle advantage of images over text for representing video content: the *codes of visual analogy* that exist between the two visual systems of icons

and video. One must be careful to understand the assertion being made here. We are not asserting a natural analogy between icons and video, but as Eco asserted for cinema itself, the existence of codes of intelligibility which support the drawing of analogies between two visual systems of representation (Eco 1976a; Eco 1976b: 191-217).

The common codes of intelligibility between the language of video and the iconic visual language used to represent video reinforce desirable properties of a representational system designed to represent video for retrieval and repurposing. An iconic visual language supports consensual description of the visually perceptible sequence-independent semantics of video. It focuses attention on those features of the visual medium of video which support its reusability by capturing salient visual properties in an analogizable form that can be compared through existing and learnable codes of visual analogy and intelligibility.

7.1.4. Arguments Against Iconic Visual Languages

We have highlighted many of the benefits of Media Streams' iconic visual language for video annotation and retrieval over and against textual systems of representation. Nevertheless, there exist several traditional arguments against the use of iconic visual languages which need to be specifically refuted:

- **not extensible**

This argument boils down to a claim that no matter how many icons one has in a system there will "never be enough of them." Clearly, this argument makes a mistake of levels. The number of icons in a composable iconic visual language is not the sum total of expressions in the language. Their valid *combinations* are the sum total of expressions in the language. If we think of letters in the alphabet or standard words in a language it is the composite forms of these primitives which provide the greatest descriptive richness.

The other point is that Media Streams' icons are embedded in a semantic hierarchy that supports extensibility through the titling of icons in the Icon Information Editor and the composing of animated icons in the Animated Icon Editor. Both of these editors make use of the fact that Media Streams has a rich enough base iconic language such that new

primitives can be meaningfully created as extensions of existing primitives.

- **not expressive enough**

In an iconic visual language for video annotation and retrieval, unrestrained expressiveness is not a virtue. Like any representation language it is designed to make some things easier and some things harder to say. The advantage of a designed controlled vocabulary is that it foregrounds those salient features of video content which can form a minimal representation for repurposing. Expressing other aspects of the content should take place in another representational system. Media Streams supports this by enabling users to attach text annotations to icons. Icons serve as consensual tokens of description which are shared among related idiosyncratic textual descriptions.

- **not easily readable**

At first glance a Media Streams screen looks like a bundle of colors and shapes, at best recognizable as a kind of computer hieroglyphics. This first impression of illegibility is not the defining one for our iconic visual language. Like a textual language, Media Streams' iconic visual language must be learned. Anecdotal evidence shows that after 20-30 minutes most people can read and write the language and that after 1-2 hours they become proficient at it. We found in our User Study (described in the next chapter) that users gain a very high degree of proficiency in the language within a few days. Interestingly, other studies have shown that pictographic scripts may actually be easier to learn for children who have difficulty learning alphabetic ones (Sampson 1985: 163).

- **not easily writable**

The anecdotal and User Study findings about the legibility of icons also apply to its writability. Nevertheless, because of the often large number of primitives (e.g., 3500+ in Media Streams), writability is an important issue to address in the design of iconic visual languages. Media Streams' iconic visual

language is a computational one that can make use of writing technologies unavailable to previous non-computational iconic visual languages. The organizational structure of the Icon Workshop makes finding and compounding iconic primitives a much more structured and simple task than in previous iconic visual languages. The Icon Query Language makes it possible to reuse the descriptive effort of others thus enabling writing to take place from dynamically configured, related palettes of icons.

7.2. Media Streams' Iconic Visual Language

The iconic visual language of Media Streams is designed to support many of the desirable features described above:

- Accurate and readable time-indexed representation of actions, expressions, and spatial relations
- Gestalt visualization of the dense, multi-layered structure of video content
- Quick recognition and browsing of content annotations
- Designed visual similarities between instances or subclasses of a class (visual resonances in the iconic language)
- Articulation of the boundaries between consensual and idiosyncratic annotations (icons can have attached textual annotations and can thus function as the explicit consensual tokens of various idiosyncratic textual descriptions)
- Global international use of annotations
- Usable by illiterate and preliterate people

Media Streams' iconic visual language is not merely a collection of visual symbols, but is, in Korfhage's sense, an iconic language as opposed to merely an iconography (Korfhage and Korfhage 1986). As an iconic visual language it has a definite semantics and syntax.

7.2.1. Icon Semantics and Syntax

Semantically, Media Streams' iconic language encompasses icons which denote both things and actions and thus embodies a distinction analogous to Chang's distinction between *object icons* and *process icons* (Chang 1986). Object icons denote objects inside the computer like a *file*. Process icons denote processes on these objects like *deleting a file*. However, Media Streams' icons have a semantic status different from icons used in traditional graphical user interfaces: the objects and processes denoted by Media Streams' icons are not computational ones, but aspects of the video content which they are designed to represent.

The semantics of Media Streams' icons are defined by the semantic hierarchies in which they are embedded (CIDIS), the Media Time Lines they are used on, and the Indices which extend the semantics of these terms through indexing of contextual examples of use.

Media Streams' iconic visual language has two major syntactic forms of organization: rules for creating compound icons from component icons and rules for creating glommed icons from compound icons. The principles of vertical and horizontal organization which enable Media Streams to form millions of compound icons from a small set (3500) of component icons were discussed in Chapter 6. The syntax of glommed icons is similar to other syntaxes for iconic sentences (Chang and others 1992; Tanimoto and Runyan 1986), Media Streams' glommed icons for actions and positions have a syntax similar to that of natural languages:



Subject - Action



Subject - Action - Object



Subject - Relative Position - Object



Subject - Action - Direction



Subject - Screen Position



Camera Motion - Object

A further form of syntactic organization is the contextualization of glommed icons within the various annotation streams of Media Time Lines. These provide context in which to read the icons. The temporal ordering of icons with their color bars also constitutes another form of syntactic organization within Media Streams' iconic visual language.

7.2.2. Future Directions

The use of icons within Media Streams often raises eyebrows and hackles (as do the more conventional uses of icons in computer systems even today (Horton 1994), yet we have found that most people who have spent some time with the system find the language easy to learn and to use and perceive the benefits of an iconic visual language for video representation.

Media Streams' current visual language is just one of many possible visual languages which could be developed to interface to the underlying ontology and representation for video content. Historically, writing systems have been developed using technologies which enabled limited use of color, motion, depth, or anti-aliasing. Shape and size proved the more consistent technological features which writing systems could make use of to differentiate signs from each other. Even in such usually imaginative science fiction as the Star Trek television programs, writing in the 23rd or 24th century has not changed much, while the writing of alien cultures is often portrayed as some variant of "Oriental" (often Chinese/Arabic/Sumerian) script, which for the West has been the signature of otherness in many domains. The striking exception to this is the *Star Trek: The Next Generation* episode, "The Ensigns of Command," in which an exceedingly alien (non-humanoid) species known as the Sheliak had a writing system so intricate that it took the Federation decades to negotiate a treaty (Star Trek: The Next Generation 1989). The Sheliak writing system (which we see for a few seconds on screen) uses various sized, colored

polygons in multiple depth planes. Our iconic visual language was designed from the outset to be written and read on a computer. As such it represents a certain evolutionary step in the development of visual languages. Writing technologies affect the development of writing systems. With the coming availability of fast, cheap 3-D graphics and video computing, the invention and use of new types of visual languages will be possible.

Future work may move in the direction of even more visually and spatially oriented forms of visual language which use the inherent arrangement of objects in a frame (and its depth planes) to convey spatial relationships rather than the sentential arrangement of icons in a glommed icon. Other possibilities include using the spatial layout of the human form to convey more information about costume and the physical characteristics of characters (a kind of dress-up doll or Mr. Potato Head approach to visual character construction and definition). Still other possibilities include using computationally parsed gesture as a writing technology for a visual language to convey motion, direction, and arrangement of characters and objects in a scene. The technological, semiotic, and aesthetic challenges and opportunities of computational visual languages will keep researchers and designers busy for centuries. What remains to be seen is the impact Media Streams' iconic visual language and future computational visual languages may have on how people communicate, learn, think, and imagine.

In the next chapter we discuss the User Study we conducted to answer key questions about Media Streams' representation and interface design.



Chapter Eight

Media Streams User Study

8. Media Streams User Study

8

8.1. Motivation

The question of the evaluation of knowledge representation systems is an open and varied one within the practice of artificial intelligence. Depending on the assumptions underlying the endeavor various standards of success are applied: formal proofs, interesting program output or behavior, the types of artifacts or processes a system enables to be constructed by people and machines, market viability, etc. In order to assess the performance and design of systems which involve theories and practices of computational and human representational activity, one must first understand the underlying assumptions about artificial intelligence which inform the systems' construction and use. Apart from the various schools and techniques of artificial intelligence research (connectionism, logic programming, frame-based systems, situated cognition, etc.), the endeavor can be divided into two strategies whose difference lies in how the relationship between computational devices and the people who build and use them is conceptualized.

In the prevailing approach, artificial intelligence is understood as the project to computationally model and simulate certain capabilities and behaviors of the human brain which we would fit under the rubric of "intelligence." Examples here include: the ability to infer new consequences from information not explicitly containing those consequences (inference and learning); the ability to recognize visual or conceptual patterns in information (perception); and the ability to understand natural language and answer questions (explanation and problem solving). This approach has sought to solve the problems of artificial intelligence by constructing an autonomous intelligent device, a kind of "Einstein in a box."

The other paradigm of artificial intelligence research does not seek to construct an autonomous computational mind, but attempts to augment human intelligence and capabilities through the interface between human beings and computational devices. We have already mentioned Haraway's paradigm of the "cyborg" that describes this model of human-computational coupling (Haraway 1991); within computer science Douglas Engelbart, inventor of the mouse and many other pioneering interface devices (Fraase 1990), championed a notion of "intelligence amplification" that may be seen as a cyborg revision of the Einstein in a box project of artificial intelligence. Examples here include: the ability to

store and access information that exceeds and augments the capacities and performance of human memory while utilizing an interface that assimilates itself to the methods of human memory; the ability to greatly accelerate the speed and precision of artifact construction through the use of computational power; and the computational augmentation of human analogical understanding through offering connections, contexts and relations for large corpora of information.

Media Streams is clearly in the tradition of augmenting and amplifying human intelligence and capabilities through intimate interfaces to computational devices. The challenge of assessing the performance of such systems within traditional artificial intelligence is that machine-centered evaluation methods are insufficient since they do not see humans as part of the cyborg system. One might think to turn to traditional usability testing methods from the discipline of human computer interaction, but these techniques are largely insufficient because they tend to focus on the evaluation of highly specific aspects of the interface itself as opposed to the functioning of the system as a whole.

One can see that the evaluation of cyborg systems is a difficult methodological task. The task was unavoidable for Media Streams because if we claim to have built a system that supports the annotation, retrieval, and repurposing of video by various users, having only one user is by definition an insufficient test.

The efficacy of the retrieval and repurposing mechanisms has been shown by offering you examples of what the system can do in Chapter 5. This evaluation method borrows from more traditional knowledge representation methodology that asks: “does the program work?” The test is to examine the program output and assess its validity. Given the output our system is able to generate we argue that the answer is yes.

The question of whether humans can use Media Streams to create a body of annotated footage that supports content-based retrieval and repurposing of video needed to be answered in a different way. From August 30, 1994, to September 1, 1994, we conducted a 2 1/2 day long user study that sought to answer very specific questions about Media Streams’ performance. These questions are not about the user interface per se, but about how humans working together with the system could construct usable, sharable representations of video content.

8.2. User Study Questions

We designed our study around answering three sets of questions which focused on the ability of humans to learn to use the system, to reuse each other's descriptive effort, and to create semantically convergent annotations of video content. We hoped that analyses of the results of our study would provide answers to the following questions:

- What is the learning curve for Media Streams? How much time spent in education and practice is necessary for new users to achieve expert-user status?
- When users have existing and relevant icons available do they make fewer new icons? What is the relationship between new icons made to existing icons used?
- To what extent do different users' annotations converge? That is, how similar are different users' descriptions of the same footage?

We also hoped that the study would provide us with the type of user-centered design critique previously unknown to our largely untested system.

The participants in the user study worked long hours over 2 1/2 days and we are very grateful to them for their participation. We also ran the study with the only Expert Users available (Brian Williams and Golan Levin) to provide a control for the New Users group.

8.3. User Study Design

In designing the study we had initial discussions with user study practitioners who advised us on guidelines and goals of human subjects study and methodologies (Debby Hindus and Karon Weber). Our greatest debt goes to the experimental psychologists and interns at Interval Research who helped us design the study and forced us to clarify the questions we were asking and the methods we were using to answer them (Diane Schiano, Ellen Bewersdorff, Herb Colston, and Ellen Levy). The study was approved by the Interval Research Corporation Human Subjects Committee. Subjects were Interval Research employee volunteers who received no additional reimbursement for their participation except for

catered breakfast, lunch, and snacks throughout the study and small Balinese painted wooden hanging sculptures as thank-you gifts at the end.

The study began with a training phase in which participants were introduced to the task. We set up the scenario that they had been recently hired by an innovative stock footage house in 1997 that had just purchased Media Streams and needed to get up to speed as quickly as possible. We used a combination of teaching by demonstration and participatory learning games to train the participants. Four workstations were used throughout the study. Sometimes participants worked in groups of two at a single workstation; other times they worked in two interleaved shifts of four persons each in which each person had a workstation to themselves. While the participants worked at their workstations we recorded their interactions with Media Streams in an automatic computer logger (built by Brian Williams) and on videotape. At the end of the third day, the participants were asked to fill out a six-page questionnaire that tapped pertinent biographical information and asked about their experiences with Media Streams. We closed the study with a wrap up discussion that was also videotaped.

The Media Streams User Study had five distinct phases spanning three consecutive days.

Day One

Phase I: Training

All participants were greeted and told that they had just been hired for a position in a stock footage house that requires them to learn Media Streams quickly. They received an introduction to the system and were given a brief demonstration of using the system to annotate a video segment. Then the group divided into four teams of two persons each for a series of learning games to gain proficiency in the iconic visual language and learn several of the basic skills needed to operate the system. The games were the treasure hunt game, the race game, and the glom game (see Appendix C for the game sheets used in the study):

- **The Treasure Hunt Game**

Participants were given a list of twenty Icon Workshop icons (such as “toaster,” “fireplace,” and “to kiss”) and told to find as many of them as possible within fifteen minutes’ time. This game was the participants’ first introduction to the Icon Workshop and bred the skills of navigating the Icon Workshop and “reading” icons.

The most successful teams were able to find 14 of the 20 required icons in the time allowed.

- **The Race Game**

Participants were given a list of ten textual descriptions of compound icons (an example was “inside a coffeehouse in San Francisco”) and told to construct as many of them as possible within fifteen minutes’ time. This game endeavored to broaden the participants’ ability to read and locate Icon Workshop icons; it also, however, introduced the new skill of combining Icon Workshop icons into compound icons.

- **Actionary — the Glom Game**

This game required each two-person team to send a member (the “A” person) to the experimenter’s station. There, the “A” people saw one or two glommed action icons on a Media Time Line. The “A” people were then asked to return to their teammates (the “B” people) and attempt to convey to the “B” people — through action only — the meaning of the Media Time Line annotations they saw. The “B” people were then required to create a log on a Media Time Line (as quickly and as accurately as possible) of the actions performed by the “A” people. The “A” people were instructed to call out to an experimenter when they felt that their “B” person had correctly reconstructed the Media Time Line; the first team to correctly produce the Media Time Line were declared to have won that round. Some of the Actionary tests that the participants were given were: “chewing,” “spinning around while talking,” and “squatting; then, patting the floor.” This exercise allowed the continued development of the participants’ icon-reading and icon-finding skills, while introducing them more deeply to the iconic Action hierarchy and the skills used in constructing glommed icons on the Media Time Line. More importantly, however, participants gained an understanding of how they might actually annotate an action. They learned the critical skills of translating actions into graphic descriptions—without the help or hindrance of intermediating words—and

creating Media Time Line annotations to convey the temporal relationships of these actions.

The winners and runners-up for each game received candy prizes. After the games, Golan Levin and Brian Williams demonstrated more system features for about twenty minutes. To end Phase I, participants spent approximately ten minutes looking at a fully annotated Doritos commercial.

The participants then broke for a catered lunch for 30 minutes. After lunch, the participants were divided into two groups of four in order to begin Phase II.

Phase II: Independent Log of Sequence 1 by Group I and Group II

Independent logging refers to the logging task performed without the availability of previously generated compound icons or glommed icons (by self or by others).

The participants independently annotated a short video sequence from the dialogue-less film noir movie, *The Thief*, by Russell Rouse (Rouse 1952). Rouse made an entire feature film without a word of dialogue using music, gaze, and the camera to tell his story. The film has the visual style of film noir and a kind of gripping intensity because of its reliance on character, actions, music, and camera to guide the viewer's attention and expectations. Participants were given the names of the main characters in the sequences (Ray Milland and Martin Gabel).

All participants were asked to return the next morning at 10:00 AM.

Day Two

The day began at 10:00 AM with a question and answer session for 30 minutes. The rest of the day until 5 PM (except for lunch) was taken up with Phase III: a series of dependent annotation sessions of four short sequences alternating between Groups I and II such that every participant annotated every sequence.

Phase III: Dependent Log of Sequences A, B, C, and D by Groups I and II

Dependent logging refers to the logging task performed with the availability of previously generated compound icons or glommed icons (by self or by others). Phase III was designed to enable us to study the reuse of descriptive effort through eight sessions of annotation.

The first session of four participants logged their sequences independently (out of necessity since they went first), while the following seven sessions were logged dependently. That is, participants inherited access to previously generated compound icons and glommed icons as the sessions progressed. Duplicate icons were eliminated using a merge-compound-icons program between each logging session. The following table illustrates how the rotation of sessions and participants attempted to control for participants' position in the sessions, order of sequence annotated, and sequence annotated.

Table 5.

	Sequence			
	A	B	C	D
1st	P1	P2	P3	P4
2nd	P5	P6	P7	P8
3rd	P2	P3	P4	P1
4th	P6	P7	P8	P5
5th	P3	P4	P1	P2
6th	P7	P8	P5	P6
7th	P4	P1	P2	P3
8th	P8	P5	P6	P7

The sequences were chosen for their brevity, variety, visual coherence, and lack of audio content. Sequence A was taken from the final montage sequence of the recent compilation film, *Rock Hudson's Home Movies* (Rappaport 1993). *Rock Hudson's Home Movies* is a fictional after-death reflection by a young actor playing Rock Hudson who by commenting on and juxtaposing scenes from Rock Hudson's various movies reveals their homosexual subtext. Part fiction, part documentary, part comedy, *Rock Hudson's Home Movies* is an outstanding example of a new type of compilation film made possible by access to a large corpus of feature films and a willingness to push the genre boundaries of the documentary and fiction film. The closing sequence juxtaposes shots from different movies of Rock Hudson and John Hall waving and performing various actions to create a kind of visual dialogue between the characters. Participants were told that the characters' names were Rock Hudson and John Hall.

Sequence B was taken from Maya Deren's silent experimental film, *At Land* (Deren 1947). Maya Deren worked within the medium of silent film long after the talkies had eliminated silent films from mass circulation. Her work on such movies as *At Land* explored the possibilities of a purely visual narrative that was not a silent film trying to talk, but the telling of stories using the tools of montage, gesture, gaze, location, and time. Participants were told that the character's name was Maya Deren.

Sequence C was taken from home video footage I shot of a hike at Mt. Rainier in Washington. During the AAAI-94 Conference this summer in Seattle a group of friends and I went to Mt. Rainier for an evening hike beginning at 6 PM and ending at 9 PM just as the last rays of the sun were fading from the sky. I had along a Sharp ViewMaster camcorder and shot footage of the hike, my friends, the mountain, flowers, and marmots. Participants were told that the characters' names were Janet Cahn and Marilyn Walker.

Sequence D was taken from Dziga Vertov's breakthrough silent documentary film, *A Man With A Movie Camera* (Vertov 1928). Vertov, another early Soviet cinema pioneer, revolutionized documentary film making with his use of montage to create mechanistic rhythms and visual effects with urban and industrial footage (Leyda 1973; Michelson 1984). Participants were told that the characters' names were unknown.

At the end of this very long day, all participant's were asked to return the next morning at 9:00 AM.

Day Three

On the third and final day of the study, participants returned to the independent annotation task so that we could have a comparison session to the first annotation session from Day One Phase II.

Phase IV: Independent Log of Sequence 2 by Group I and Group II

Independent logging refers to the logging task performed without the availability of previously generated compound icons or glommed icons (by self or by others).

In this phase, the participants independently annotated a second similar sequence from *The Thief*, by Russell Rouse (Rouse 1952).

Phase V: Debrief

After both Groups finished, all participants joined for a catered sushi lunch to fill out Media Streams User Study Exit Questionnaires and have a videotaped wrap-up discussion.

For more detailed documentation about the user study we have supplied the following appendices: Appendix C for the games we used in the Media Streams User Study; Appendix D for the Media Streams User Study Exit Questionnaire; Appendix E for the Media Streams User Study Exit Questionnaire numerical results and user comments for the new users and the expert users; and Appendix F for the transcript of the Media Streams User Study Wrap-Up Discussion.

8.4. User Study Participants

The experts for our Expert User control study were my two co-designers of Media Streams: Brian Williams (age 23) and Golan Levin (age 22). The participants for our New User study were employees (and one spouse of an employee who graciously filled in because of a last minute absentee) from Interval Research Corporation who work in a project in which they are involved in shooting and logging ethnographic style documentary footage. For the sake of discussion I have changed the names of our New User study participants. They were:

NAME	AGE	SEX
Joshua	23	M
Sarah	24	F
Raphael	27	M
Vladimir	32	M
Jane	29	F
Erin	36	F
Sandra	40	F
Betsy	50	F

In the Media Streams User Study Exit Questionnaire, the participants in the New User and Expert User studies described their prior relevant experience and their experiences of using the system. The full text of the Exit Questionnaire may be found in *Appendix D: Media Streams User Study Exit Questionnaire*. The complete numerical results and user comments for the New Users and Expert Users may be found in *Appendix E: Media Streams Study Exit Questionnaire Results*.

As far as relevant education: Joshua and Jane attended film school and studied film theory; Sarah studied studio art; Raphael attended film school, art school, and studied film theory; Sandra attended art school and studied film theory; and Betsy studied film theory. The New Users' television viewing ranges from 0 to 15 hours per week with a group average of 9 hours. They watch from 0.5 to 4.5 movies per week with a group average of 1.91 movies. Among the Expert Users, Golan attended art school and studied film theory; Brian has taken an introductory film analysis class. Golan watches 10 hours of television and 3 movies per week. Brian watches 14 hours of television and also 3 movies per week. On average, the Expert Users are about 10 years younger, watch one and a third times more television, and see one and a half times more movies than the participants in the New User study.

Both New Users and Expert Users also described other task related prior experience. The average results for each group are rated on a scale from 1 (none) to 7 (lots) and compared in the following chart:

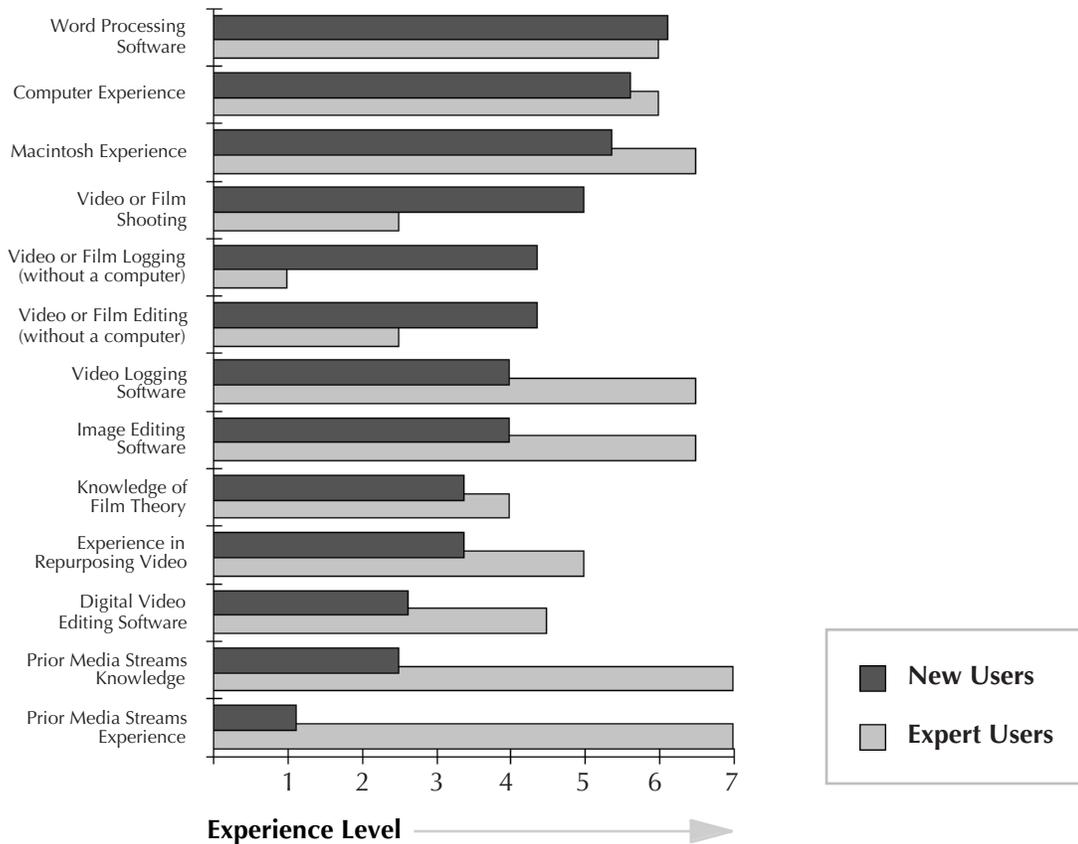


Figure 88. Relevant Prior Experience

The most significant and largest difference in relevant prior experience between the New Users and the Expert Users is, as expected, their familiarity with Media Streams itself. The point spread between Expert Users and New Users in terms of *Experience with Media Streams* was **5.88** and in terms of *Knowledge of Media Streams* was **4.50**. Interestingly, the New Users on average possessed certain prior relevant experience that the Expert Users largely lacked: hands on experience with traditional video or film production and logging. The point spread between New Users and Expert Users in terms of experience with *video or film logging without a computer* was **3.38** and in terms of experience with *video or film shooting* was **2.50**.

8.5. User Study Results

The user study was ambitious and exhausting for experimenters and participants alike. It demanded a tremendous time commitment in order to simulate 2.5 days of active work using Media Streams. It took three months of preparation to design, plan, and implement the study much of which involved transforming Media Streams from a demo into a usable tool that could withstand eight new users over three days. Of course we did not have actual conditions of use to study since the system is a research prototype, but the work of the participants did enable us to obtain results which point toward answers to the questions we set out to investigate.

To summarize our results, we did answer our three questions in the affirmative: the system is learnable; when previously existing and relevant icons were available, users made fewer new icons; and different users' descriptions of the same footage are more similar to each other than different users' descriptions of different footage. The results for each question are presented in graphs and explained below.

8.5.1. Improving Annotation Rate

What is the learning curve for Media Streams? How much time spent in education and practice is necessary for new users to achieve near expert-user status?

The first graph below shows how the New Users' performance improved from Day One Phase II to Day Three Phase IV compared to expert level annotation performance on the same movies. The X axis corresponds to an annotation session of 62 minutes. The Y axis corresponds to the average number of total annotations made at that point in the annotation session. The average expert annotation rate is represented by the top line. The average Day One Phase II New User annotation rate is represented by the bottom line. The average Day Three Phase IV New User annotation rate is represented by the middle line. The expert annotation rate was 1.08 annotations per minute. After two hours of training, New Users started at Day One Phase II with an average annotation rate of .35 annotations per minute and by Day Three Phase IV had reached an average annotation rate of .65 annotations per minute.

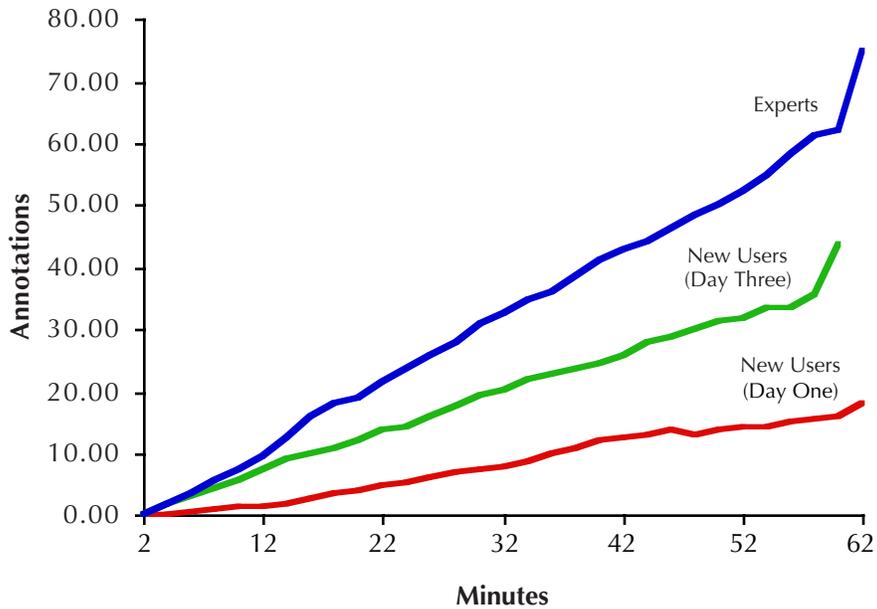


Figure 89. Annotation Performance (Experts and New Users).

In the wrap-up discussion, many participants felt that they could become expert users in two to three weeks.

Marc: How long do you think it would take to become a total whiz on the system?

Joshua: Two or three weeks of days like this.

Even though this assessment may be somewhat optimistic on the part of participants, anecdotal evidence corroborates the rapid learnability of the system. In showing people the system, I find that within 20-40 minutes, most people understand the basics of the language and the interface. Within 60-90 minutes they are able to annotate video. In a recent session with a 13 year old male, he was able to use the language and the interface, annotate, retrieve, and repurpose video within a few hours. The initial shock of seeing a screen full of icons is not a good indicator of the difficulty of learning the system. In fact, it seems to be a counter-indicator, since the system seems quickly learnable, internally consistent and reinforcing of its design principles, and able to be mastered in a relatively short amount of time.

The Exit Questionnaires also provided information about what tasks New Users and Expert Users found more or less difficult. The graph below illustrates the differences in perceived difficulty for particular tasks in Media Streams between New Users and Expert Users on a scale of 1 (easy) to 7 (hard). The Y axis lists the various tasks (sorted in order from most

difficult to least difficult for New Users) and the X axis represents the perceived difficulty:

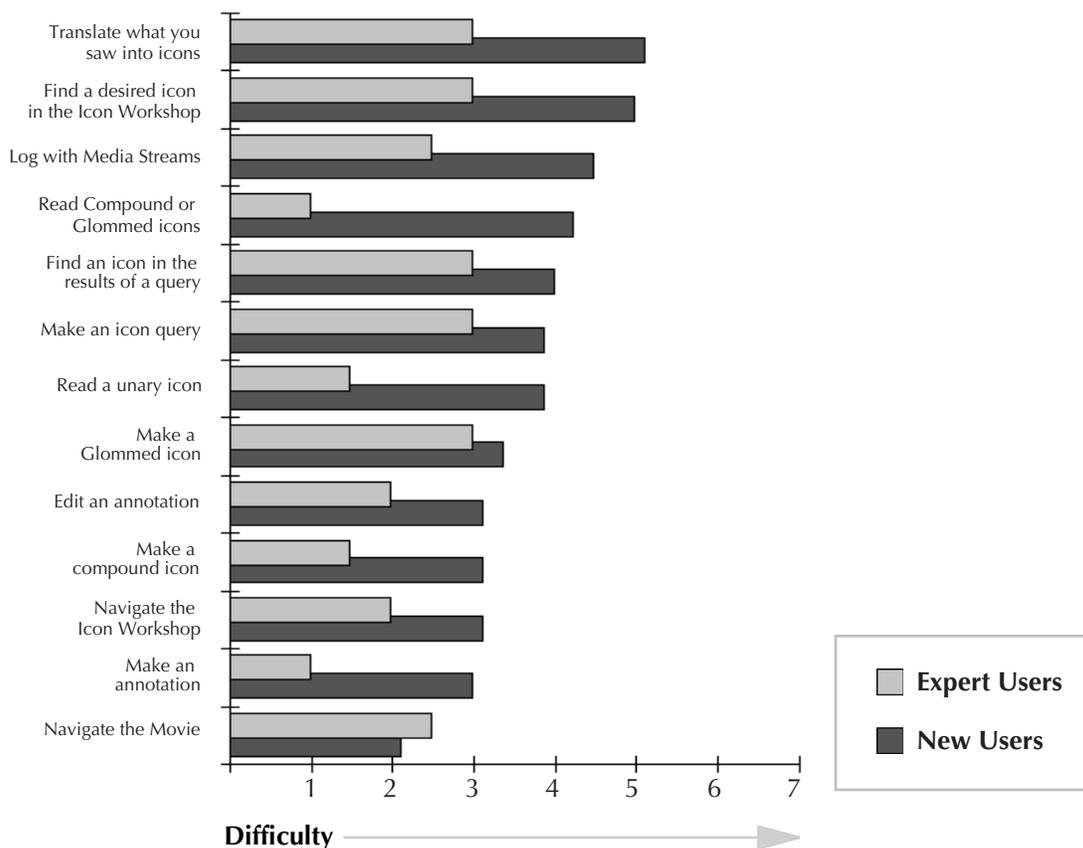


Figure 90. Perceived Difficulty of Media Streams Tasks

The greatest differences between Expert Users and New Users had to do with their ability to read and use Media Streams’ iconic visual language. The tasks with the most divergent point spreads between New Users and Expert Users were:

Table 7.

Media Streams Task	New Users	Expert Users	Delta
Read Compound or Glommed Icons	4.25	1.00	3.25
Read a unary Icon	3.88	1.50	2.38
Translate what you saw into icons	5.13	3.00	2.13
Find a desired Icon in the Workshop	5.00	3.00	2.00
Make an Annotation	3.00	1.00	2.00
Log with Media Streams	4.50	2.50	2.00

Surprisingly, in the task of *Navigating the Movie* New Users (2.13) found it easier than Expert Users (2.50). This is most likely attributable to the New Users' experience of the novelty and comparative ease of navigating video digitally as well as to the Expert Users' expectations of what the system ideally should be able to do.

Significantly, the top four most difficult tasks for New Users are tasks which Expert Users find markedly easier. Not only can users learn Media Streams, they can improve their performance significantly on those tasks they initially find the most difficult.

8.5.2. Reuse of Descriptive Effort

When users have existing and relevant icons available do they make fewer new icons? What is the relationship between new icons made to existing icons used?

We also answered the questions about whether users would use existing relevant icons vs. making new ones. The trend is what we expected and bodes well for the possibility of annotators using each other's work. This will allow the person annotating the fifty-first episode of *Gilligan's Island* to do so more quickly and with greater consistency than the person logging the first. In the graphs below, the X axis represents the respective sessions of dependent logging from Day Two Phase III. The Y axis represents the total number of new icons made. In the first graph below we see the number of new icons created by each annotator from Session 1 to Session 8 in Day Two Phase III. In this part of the study, each successive annotation session had access to the icons created in the sessions before it.

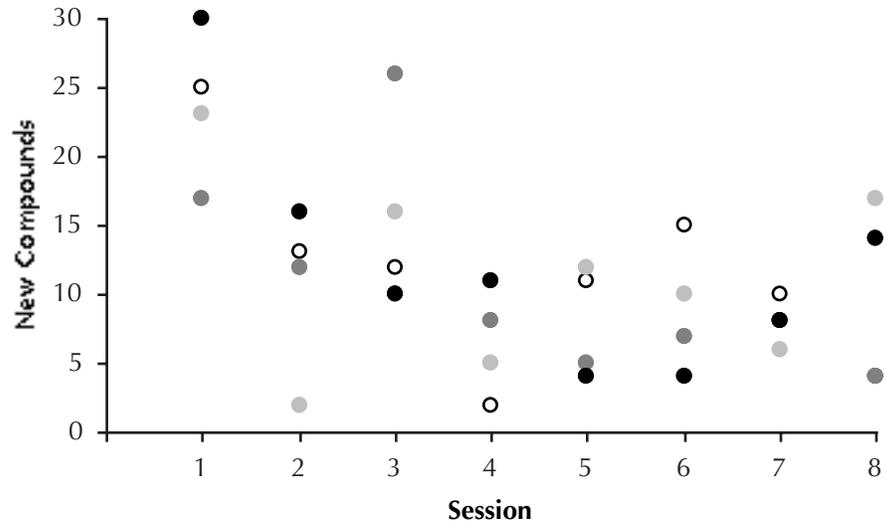


Figure 91. New Compounds Created Per Logger Per Session

In the second graph we see the normalized (one data point was thrown out because the participant was late and only logged for 5 minutes as opposed to 60 minutes) averaged trend line of new icons made per session that shows us the result that as more relevant icons became available the number of new icons created decreased such that by the eighth session 2.5 times fewer new icons were created than when annotating the same footage without existing icons available.

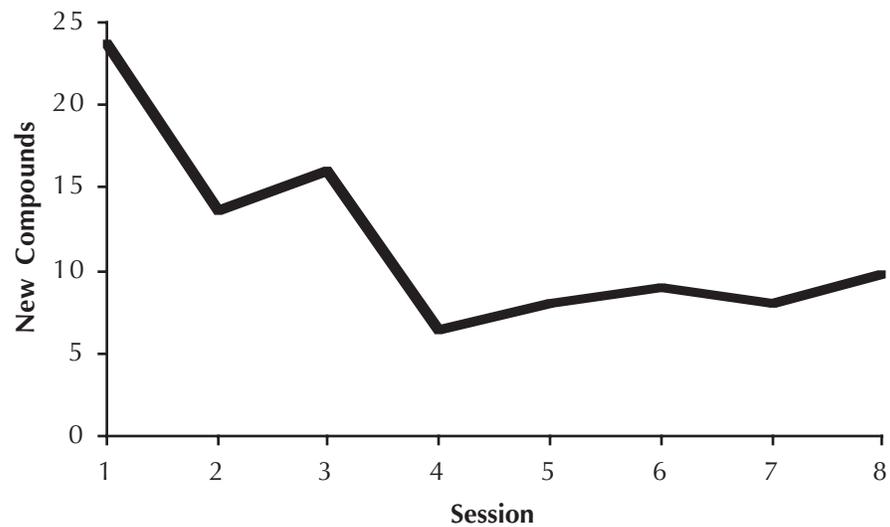


Figure 92. Normalized Average New Compounds Created Per Session

In the wrap up discussion, participants had different experiences of having access to and using each others' icons.

Golan: How helpful was it to have other people's icons?

Sarah: I personally found it kind of disturbing because — well I don't know if it just has to do with learning the ontology like she [Erin] was saying — but I would look for “someone has a hat on their head” and someone would have put in “someone has a hat on their hair,” and I was like, should I use this or should I just make another one? There were a lot of things like that where they were there, but then there was this issue of whether I should use them or not.

Joshua: I used those as a shortcut by bringing them up to the Workshop, and you'd get to something near what you wanted.

Sarah: That's true.

Vladimir: I found it very useful actually because you could find pretty much a lot of the icons that you actually need: Just bring them up there instead of going and searching for them through all the hierarchies, and then use them to build your own. I also found it useful in making a first pass and describing the main events. Then later, if I had time I could go back in and just add stuff.

Though participants had various styles of and attitudes toward appropriation and use of other people's icons, the overall trend of the group was towards significant reuse of descriptive effort. The data and the users' comments point toward two interesting questions worthy of further study: how much does the “quality” of the first icons created affect the creation of new icons when users are annotating similar footage; and does the reuse of descriptive effort reach a certain plateau, i.e., is there a consistent level of new icon creation inherent in people's annotation practice even when all relevant icons are available? The Media Streams' User Study and the Media Streams system now enable us to ask these kinds of pertinent and important questions about shared annotation systems which enable the creation of cyborg memories and reusable repositories of representational activity.

8.5.3. Convergence of Descriptions

To what extent do different users' annotations converge? That is, how similar are different users' descriptions of the same footage?

The participants in the Media Streams User Study did not perform retrieval on the footage they annotated. In order to test the convergence of their annotations, we ran their annotations through the same algorithm we use for retrieval by comparing each user's annotations of the shots of one sequence to other users' annotations of the shots of the same and of other sequences. The sequences and logs used are from Day Two Phase III of the study.

The points in the graph below represent a measure of similarity between two different users' logs. The semantic dissimilarity axis (X axis) represents the semantic distance between the annotations of the two compared logs within the Icon Hierarchy, while the incommensurability axis (Y axis) represents the ratio of actual annotation matches to possible matches between the two logs being compared. The (0,0) point on the graph would represent the position of a comparison between absolutely identical logs (logs in which *all* annotations are semantically identical). Comparisons of two logs of the same movie are represented by an X; comparisons of logs of different movies are represented by an O.

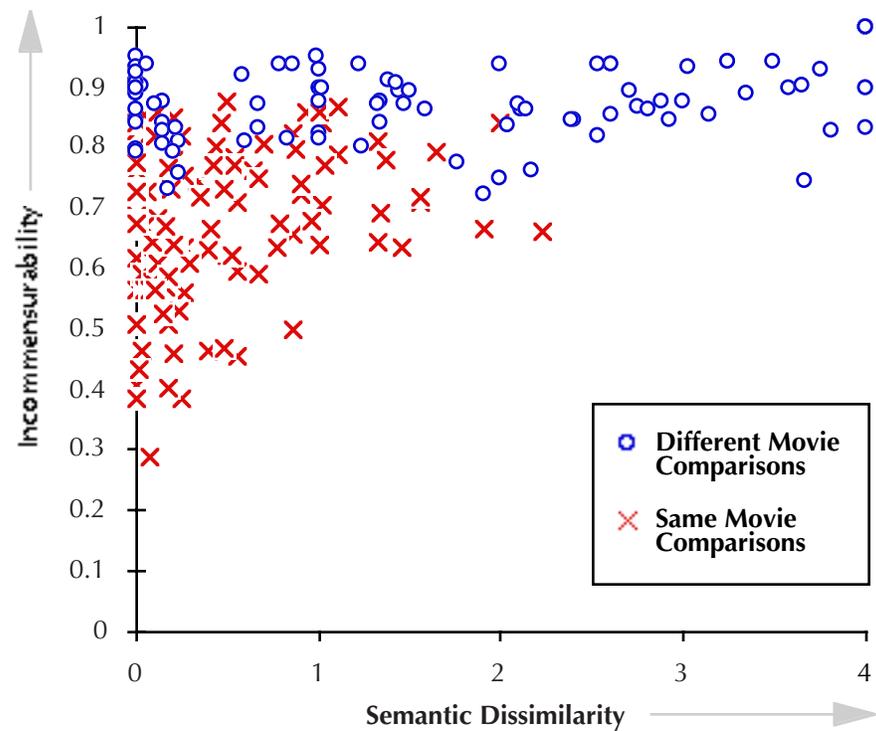


Figure 93. Comparison of New User's Logs

The analysis of the two populations (comparisons of the logs of different movies, and comparisons of logs of the same movie) revealed statistically significant differences between the two groups along both the semantic and incommensurability axes. *Participants were a random factor.* Along the axis of semantic dissimilarity, logs of the same movie by different participants were significantly more similar (mean = .4675) than logs of different movies (mean = 1.802), $t(9) = 10.22$, $p < .0005$. Likewise, along the axis of incommensurability, logs of the same movie were significantly more similar (mean = .6754) than logs of different movies (mean = .8921), $t(100) = 15.421$, $p < .0005$.

The distribution along the semantic dissimilarity and incommensurability axes of the same movie comparisons conforms to our hypothesis, since if the annotator is not a dependent variable, logs of the same movie by different annotators would tend toward semantic identity. Though these results were obtained using a small population of annotators (8) over a small corpus of footage (4 sequences), they are statistically significant. It is also very important, since it indicates that different annotators can use Media Streams to create annotations which can be used by people other than those who created them for content-based video retrieval and repurposing. Unlike keyword-based and natural language video annotation systems, Media Streams enables different users to create semantically convergent representations of video content.

The Expert Users' logs exhibited a similar pattern of semantic convergence. Although there were too few data points from the Expert Users to get meaningful statistical results, the similarity of the Expert User distributions to those of the New Users is corroborative of the general trend and promising.

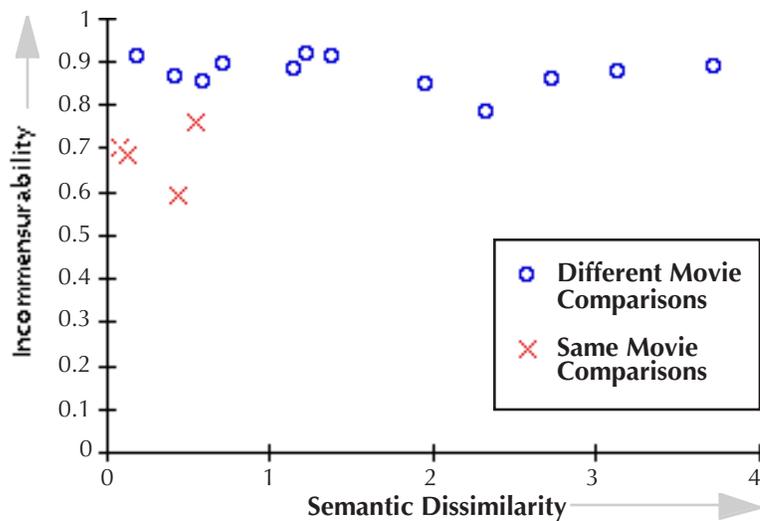


Figure 94. Comparison of Expert User's logs

In the wrap-up discussion participants speculated about how they might be able to use the logs they created for retrieval and repurposing. Jane, a very experienced documentary filmmaker who had the most resistance to and trouble with logging with the iconic visual language, discussed using the system for retrieval and repurposing with Joshua, who enjoyed and felt mastery when using Media Streams, and who engages in a film making practice that reuses pop cultural materials and as such is less “cinematic” than the type of film making Jane does:

Joshua: As far as repurposing, which I know something about, I actually know personally — maybe I'm just not thinking far enough into the future — but I can't personally think of making, as something I'd want to do, making something happen made up in the explicit way you're talking about, just made up of random other pieces. They're important because they have cultural significance attached to them, and it matters whose hand it is, and it matters where it came from, to me. And I don't start something with an idea, that well I want this to happen, I'm going to find the footage to do it; it's almost the reverse: here's the footage that I've seen, these things are memorable, I've kept them, I'm going to go and reuse them. I'm still wrestling with that because one way that I thought would be really interesting is having all these truck and pan motions and stuff, that would be great to use because I love the idea of decontextualizing that. I think you could do something really fun with assigning different values of [e]motions to different kind of sounds.

Jane: But just think, if you were trying to do a montage and you've got someone diving off of a diving board. So you want all the shots of people jumping off a diving board and then you are gonna make it into one big dive.

Joshua: See but I don't think that way. I mean I can see how I would, and how it would be useful.

Marc: Jane, could you elaborate?

Jane: Well, I could see where you could use a system like this for something like that. You could just put in an icon, and maybe you don't care if it's a man or a woman who's diving. [...]

Jane independently reproduced the central example in this thesis (before it was the central example) and offers a notion of repurposing which relies on decontextualized properties of the video content. Joshua engages in a highly personalized, intertextual practice of repurposing which uses different principles of construction than the more traditional narrative forms of continuity Jane practices. Media Streams can support both modes of construction. For Jane, the fit is simple and intelligible in her current practice of video sequence construction, though new continuity forms would arise in reusing stock footage. For Joshua, extending the representation to capture some of the pop cultural specificity that he desires would make the tool more useful for him (in the study he was not made aware of the ability to create user definable text annotations of icons that would support his descriptive practice though in a non-standardized way). Media Streams does offer him a level of control over content that would enable him to experiment with some of the new formal organizations he proposes as well as others yet unimagined. Joshua also assented that although the system did not capture the cultural resonances and traces he is interested in, the annotations he created with Media Streams' minimal representation of video content could be used by other people than himself:

Joshua: [...] most of the time I managed to say a bare-level thing that you could say, "this is going on, that is going on, this is going on." It's enough that if you needed to go look at it you could see what else is there.

Marc: Do you think others would be able to use that log?

Joshua: Mm-Hmm.

The mass of data collected during the 2 1/2 days of the study and the observations and experiences of the study participants will have an impact on the development of Media Streams for years to come. In addition to validating our three hypotheses—that Media Streams is learnable, that users will make fewer new icons when relevant preexisting icons are available, and that different users' annotations of the same footage will be semantically convergent and of different footage will be semantically divergent—the User Study offered some specific and revealing insights about the representation of video content and the interface to those representations in Media Streams.

8.6. Insights from the User Study

The participants offered many insights and suggestions about Media Streams (for a complete transcript of their written and spoken responses please refer to the Appendices). In the following sections we focus on a set of ideas the participants in the User Study generally and strongly agreed on about how to improve the system.

8.6.1. Interface Issues

Most of the participants asked for a way to search for and replace a given component icon in a set of compound icons used in a log on a Media Time Line. For example, this facility would enable a user to change all or selected instances of “*Adult male waving at adult female*” to “*Rock Hudson waving at adult female.*” While we had realized before the User Study that such a facility would be useful, the feedback of the participants will result in its implementation in the next version of the system.

Another feature most of the participants asked for was a “Find Icon” interface for quickly locating an icon in the Icon Workshop by typing in a textual description. We had been developing such an interface in the months leading up to the User Study with Anil Chakravarthy, who was interning at Interval Research Corporation to work on this and other projects. The interface remains to be completed, but will enable users to quickly find component icons and compound icons in the Icon Space by typing in text and then selecting from a set of returned icon choices.

It is very important to note that the participants requested this interface not so that they could annotate with text or read textual logs, but to speed up the admittedly cumbersome process for New Users of navigating the Icon Workshop to quickly locate a desired icon. Expert Users have much less trouble doing this since they know the make-up of the Icon Hierarchy in greater detail and they are more adept at the Icon Query Language. The Icon Query Language leverages the search effort by enabling users not just to find single icons, but *related sets of icons*. When one is looking for one icon, a textual Find Icon interface could potentially cut through the Icon Hierarchy with greater speed than the current point and click interface. In the case where experienced users have access to and are adept at retrieving useful sets of iconic descriptors, the textual Find Icon interface would probably be less helpful to them than to the New Users in our study who had access to a small set of relevant preexisting icons and limited knowledge of the Icon Query Language. In the User Study, the Icon Query Language was not fully available (we disabled the temporal-overlap linker  and was not a major focus of the training. One of the Expert Users

commented in his Exit Questionnaire that one of the things he found hard was:

Golan: Working with one hand tied behind my back: not having access to certain of the system's functionalities, such as the co-temporaneous linker.

The desire for a textual Find Icon interface reveals one of the issues in *writing* an iconic visual language with a sufficiently large number of iconic primitives (think of the related issues involved in designing Kanji typewriters and word processors). The Find Icon interface should help alleviate some of the navigational tedium for new users as does the existence of a rich set of relevant preexisting icons and greater familiarity with the Icon Query Language for more experienced users sharing a common reusable corpus of descriptive effort.

The asymmetry between the effort of reading vs. writing an iconic visual language is well known (Sampson 1985). What was illuminating from the user study in this regard was support for anecdotal evidence that users quickly learn to "read" the iconic visual language directly rather than translating it first into a natural language representation. In early experiences of having people use the system, two members of the MIT Media Lab's Interactive Cinema Group used an early version of Media Streams to annotate the crop dusting sequence from Alfred Hitchcock's film, *North by Northwest* (Hitchcock 1959). At first, one of the users was insistent on using the text help balloons to explain the meaning of the icons in words. After about an hour of using the system, he was no longer availing himself of the textual help balloons. When I asked him why, he replied that now he "thought the picture." This apparent cognitive shift is corroborated in the participants' comments in the wrap up discussion in the user study:

Sandra: [...] I found myself looking at the compound icons as a meaningful unit. Instead of reading it, it was enough to just look at it and not translate it into words.

Further study may reveal to what extent the ability to "think the picture" facilitates the annotation, retrieval, and repurposing of video content across different communities of users, literate, aliterate, and preliterate.

8.6.2. Representation Issues

The wrap-up discussion revealed some important tensions in the representational design of Media Streams that may not have come to light without the benefit of the participants' work. The first issue that concerned many participants was the inability to use spatial locations as objects of relative positions or actions. The problem here is one of a changing relationship between figure and ground. In many cases, what was the background in a scene (its spatial location) becomes the focus of an action or relative position of a character. For example, imagine we describe a scene as "located in front of an institutional building" and then the main character turns to look at the institutional building. What was the location is now the object of a character action. If we add to the objects annotation stream an icon for the object of the institutional building a potential ambiguity arises as to the relationship between the institutional building as object and the institutional building as spatial location—are they the same or different? Media Streams does not currently support taking part of a spatial location and using it as an object. We will be modifying the representation to support this operation that would allow objects in the background (spatial location) to be unambiguously used as objects in actions and relative positions.

The participants expressed some difficulty with using screen position icons to indicate direction of action in glommed icons. The problem is a deep one. The visual vocabulary needs to do a better job of making clear what coordinate system a description of direction or position is in: body-centered (using the position and orientation of a character or object as the origin) or screen-centered (using the viewer's position and orientation as the origin). This issue will be addressed in the next version of Media Streams.

The final representational design issue that arose in the User Study harkens back to Abelson and Sussman's requirement that a programming language have a "means of abstraction, by which compound objects can be named and manipulated as units" (Abelson and others 1985: 4). Media Streams does not currently have a designed means of abstraction: the naming of a group of expressions as one new expression. We do support titling of icons, but that is only a means of extension of the vocabulary of primitives. What Abelson and Sussman describe is what would enable Jane to describe a pattern of annotations as "a shot reverse shot" and use that new abstraction to describe another instance of a shot reverse shot. Designing a workable means of abstraction is a challenging and important problem for video representation systems. It will also be addressed in future research on Media Streams.

8.7. Learning to See in a New Way

The User Study confirmed many of our hypotheses about the efficacy of our approach to video representation for retrieval and repurposing. It also provided a set of new questions and issues to investigate. What remains most apparent from the study is that Media Streams, like all representational systems, embodies a particular way of seeing the world (Brachman and Levesque 1985; Heidegger 1980; Winograd and Flores 1986). It seeks to instill a practice of description that supports the creation of reusable video. The task of annotating video in Media Streams for repurposing requires users to rethink their most basic assumptions about how to describe locations, times, actions, characters, and events. The ability to look at video content and describe it such that it can be used in the widest variety of new contexts demands that users decontextualize the inferences they make about what they see. As Scott McCloud makes clear in the following comic, supporting the creation of new inferences through decontextualization makes room for the unexpected (McCloud 1993: 61).

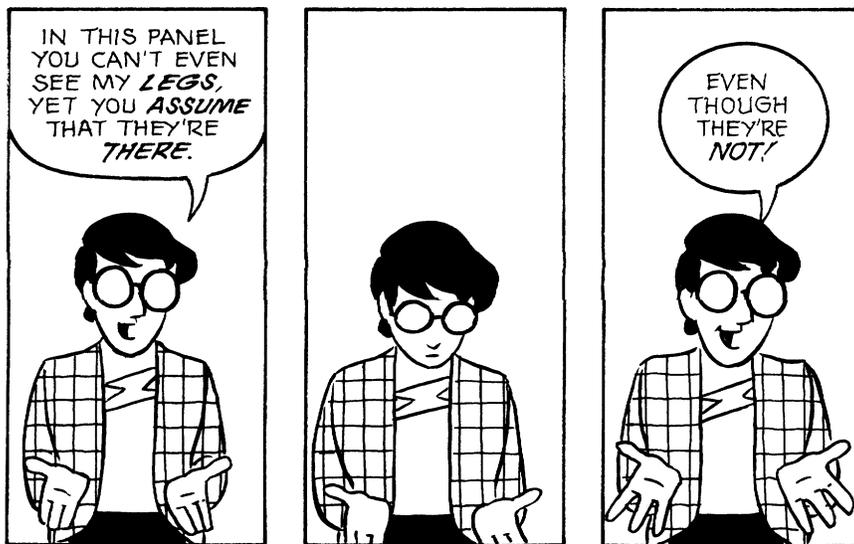


Figure 95.

The Russian Formalists argued that art's function is to "defamiliarize the familiar" (Stam and others 1992: 10-11). Paradoxically, by defamiliarizing our acts of aesthetic cognition through the description of video in Media Streams, we transform representational artifacts into raw materials which support new acts of defamiliarization.

Yet this new way of seeing is learnable, even in a few days, as Vladimir describes in his understanding of the advantage of representing video with Media Streams:

Vladimir: But that's the advantage actually of having pictures because also the interpretation after that is up to the person who's reading it. So I said well "taking a picture": he's using his hands, there is a camera, so there is him, there's his hands and a camera. So when we look at this, it's clear that he's taking a picture.. or at least that he's holding a camera... especially when there is the motion.

By offering a new way of looking at video, Media Streams instrumentalizes the reframing of perception and interpretation conceptualized in formalist and reception aesthetics. We transform reader-response and formalist film theory into an analytical-compositional constructionist practice: a *techne* of *theoria*.



Chapter Nine

Related Work

9. Related Work

9

Throughout this thesis I have made use of and made reference to relevant work from a variety of disciplines ranging from knowledge representation, to film theory, to the history and theory of visual communications systems. In this chapter I will extend this discussion by examining in more detail foundational and recent work in knowledge representation for video and by outlining the contributions and shortcomings of the very few examples of integrative research, which like Media Streams, attempt to combine a wide spectrum of technologies and methodologies to solve the problems of representing video for retrieval and repurposing.

Until the mid 1980's very little research on computational technologies and methodologies for video representation occurred within the discipline of artificial intelligence or film theory. Both disciplines faced three challenges which made this research very problematic until now:

- lack of necessary technology
- lack of motivating funding sources
- lack of cross-disciplinary expertise

Until very recently, conducting research on video representation required the mastery of arcane analog technologies like videotape and laserdisc. The availability of inexpensive, computationally parsable and combinable digital video enabled both an explosion of creative, low-budget movie making and the possibility of rapid prototyping of representations and interfaces for video manipulation. Our own work on Media Streams began with laserdisc technology but at the earliest opportunity connected Macintosh Common Lisp (Apple Computer 1993a) to Apple's QuickTime digital video technology (Apple Computer 1993b). With the advent of QuickTime, what we saw in our own work and in the work of others at the Media Laboratory and elsewhere was a new infusion of activity and a new ease of manipulation in the research area of video representation. We anticipate that this technological change, and the technological challenges it introduces, will change the field of video representation from one of a few practitioners to a major subfield in artificial intelligence and film theory.

As important as the change in the enabling technology for research in video representation is the current change in the funding matrix of the discipline of artificial intelligence that began with the end of the Cold War in the late 1980's. Artificial intelligence has had an intimate and long term relationship with the funding agencies of the military-industrial complex (first ARPA, then DARPA, now ARPA again). This relationship has privileged and promoted certain research agenda over others. With the dwindling of military funding, artificial intelligence researchers are seeking economic support from new industrial sectors: the converging entertainment and telecommunications industries. As Joseph Bates made clear in his 1994 AAAI speech, the discipline must turn to new sources of funding and consequently, new areas of research which he dubs "interactive entertainment" (Bates 1994a). Warren Sack and I have made this point in relationship to artificial intelligence research about television (Sack and Davis 1994). With this change in funding will come a change in the questions artificial intelligence researchers deem worthy of pursuing. Various workshops at AAAI over the past few years have functioned as seedbeds and harbingers of these disciplinary changes: AI and Interactive Entertainment (1990, 1992, 1994), Intelligent Multimedia Interfaces (1991), Indexing and Reuse in Multimedia Systems (1994). This excerpt from Joseph Bates' call for "technical papers on AI, the Arts, and Entertainment" for the Twelfth National Conference on Artificial Intelligence (AAAI-94) reflects this decisive disciplinary (and economic) shift (Bates 1994b):

The American Association for Artificial Intelligence has decided that the 1994 National AI conference should be different than it sometimes has been perceived in recent years. AAAI-94 is intended to emphasize new, exciting, innovative, and controversial research. The reviewing process has been changed significantly to recognize this broader spectrum of research.

One aspect of this change is the welcoming of technical papers on AI, the Arts, and Entertainment. I will oversee reviews of papers in this area, and I would like to encourage those of you working on this and related themes to submit papers to AAAI-94.

The area is potentially broad, and includes basic and applied study of AI and related technologies (such as artificial life, neural networks, robotics, and genetic algorithms) in areas such as:

- Film and video production
- Computer graphics and animation
- Interactive art (in any medium)
- Interactive fiction and role playing games
- Simulated worlds, virtual reality, video games
- Autonomous agents
- Believable interactive characters
- Music, sound, and speech
- Drama and story-telling
- Robotics, animatronics, toys
- Theme park applications

Since its inception, the Media Laboratory has conducted its research within this predominately post-Cold War funding structure that looks to the entertainment and telecommunications industries for support. Consequently, our research could be supported from its beginning in an economic-institutional framework (with its attendant priorities and issues) that the discipline of artificial intelligence is just beginning to orient itself toward.

Finally, the third reason for the relative scarcity of research on video representation stems from the necessarily interdisciplinary nature of the research. Just as the early Soviet film pioneers combined engineering and art, current research on video representation requires an admixture of disciplinary backgrounds, methodologies, and expertise. This practice of disciplinary alchemy in the service of the creation of new technologies in a new area of research has its roots in the Media Laboratory and grew within the Narrative Intelligence Reading Group; it is beginning to crop up within existing and new academic departments around the world which seek to combine computer science and media studies into a practice of artifact and theory construction. What remains to be seen is the extent to which this interdisciplinary practice will not just change computer science departments (whose funding pressures may demand it), but change the curricula and research practices within media studies, film theory, and the humanities.

Although the conditions which might foster research on video representation were not present until recently, pioneering work was done in the early 1980's by certain knowledge representation researchers who laid the foundations and helped chart the directions for our current work.

9.1. Knowledge Representation Approaches

9.1.1. Bloch: A First Attempt at Video Representation

The most important prior work done in artificial intelligence and video representation is the research of Gilles Bloch (Bloch 1987). In his short unpublished paper he outlines the issues involved in applying Roger Schank's conceptual dependency representation for action to representing and resequencing video segments (Bloch was a student of Roger Schank while at the Yale University Department of Computer Science). Bloch also discusses using Noël Burch's categories for transitions, and mentions the importance of gaze vectors in video (Burch 1969). His prototype system supposedly was able to construct simple video sequences according to Schankian scripts using conceptual dependency primitives. Unfortunately, Bloch's Ph.D. thesis is untranslated and my French is too rudimentary to make the effort worthwhile. From what I have gleaned from those who have read it, he covers important intellectual ground for video representation, though the extent to which he deals with the issues of representation for repurposing is unclear. I also believe that he left the issue of how representations are written and used unaddressed. This issue is crucial for creating a usable knowledge representation for video. Unfortunately, Bloch's untimely death cut off his fruitful early path of research in applying artificial intelligence techniques to the problems of video representation. His pioneering work began the process that has led to the discipline of artificial intelligence beginning to seriously look at the creation of representations for reusing visual media. His research, based on Metzian film theory (Metz 1974), also captured some important features that structure attention and story in video: namely gaze, action, and direction of movement.

9.1.2. Schank and His Students: From CD to CBR

The work of Roger Schank and his students is a major source of precedent and inspiration for current work in action representation, story understanding and memory-based representation. Schank's work covers many areas and has evolved through several distinct phases. The aspects of his work especially relevant to my research are his research on conceptual dependency, dynamic memory, and the case-based reasoning work that grew out of it.

9.1.2.1. Conceptual Dependency

Conceptual dependency reduced all of human action to a small set of composable primitives (Schank and Rieger 1985: 124):

The twelve ACTs are:

ATRANS	The transfer of an abstract relationship such as possession, ownership, or control.
PTRANS	The transfer of physical location of an object.
PROPEL	The application of a physical force to an object.
MOVE	The movement of a bodypart of an animal.
GRASP	The grasping of an object by an actor.
INGEST	The taking in of an object by an actor.
EXPEL	The expulsion from the body of an animal into the world.
MTRANS	The transfer of mental information between animals or within an animal. We partition memory into CP (conscious processor), LTM (long-term memory), and sense organs. MTRANS takes place between these mental locations.
CONC	The conceptualizing or thinking about an idea by an animal.
MBUILD	The construction by an animal of new information from old information.
ATTEND	The action of directing a sense organ towards an object.
SPEAK	The action of producing sounds from the mouth.

This work has a certain appeal for its rigor and simplicity, yet it has an apparent deficit for application to video representation: the semantics of human action within video are not fixed and change on recombination. However, certain of Schank's ACTs do work for video representation in much the same ways as our own action representation. It is illustrative to examine which do so and why. In video sequences, what is invisible or unheard does not exist. Mental states, intentions, and acts of cognition are all inferred to exist by viewers using the cues provided by purely physically-based actions in video. Therefore, out of the twelve ACTs, all of

those having to do with mental or invisible actions—ATRANS, MTRANS, CONC, and MBUILD—would not work as primitives in an action representation for repurposable video content. That is not to say that video sequences cannot occasion the inferring of the presence of these actions by viewers; it is simply that these actions are not depicted in the video content. Out of the eight remaining ACTs, only four of the physically-based ACTs are unproblematic: PTRANS, MOVE, GRASP, and EXPEL. INGEST has the possibility of a subtle uncertainty due to the different visual properties of interior and exterior spaces. The interior of the body of an actor is usually invisible, hence the result of an INGEST action may not be visible. Certain magic tricks play with this visual uncertainty: I appear to swallow an egg, but look, it's behind your ear! SPEAK is somewhat complicated by the ventriloquist possibilities of looping, dubbing, voice-overs and off-screen speech. An actor may be moving his or her mouth, but the sounds one hears may not be inferable as emanating from the actor's mouth. PROPEL is ambiguous due to the issue of determining the origin of a motion. Multiple object actions have an indeterminate causality in video that is related to question of animacy. For example, if I have a video of a bat swinging, contacting a ball, and the ball flying away from the bat, I could say: a bat PROPELs a ball. However, by doing so I am reducing the indeterminacy of this sequence of actions to one possibility by making an interpretation of the origin of the motion when it is actually ambiguous: Did the ball run away from the bat or did the bat hit the ball? The last remaining ACT, ATTEND, is also problematic. To the extent that "the action of directing a sense organ towards an object" can be visually observed, ATTEND can be used to describe an action in a video sequence. But the motion of a head or eyes toward an object does not necessarily imply seeing: external visual actions do not necessarily entail internal cognitive actions in video.

Of course, the twelve ACTs were not designed to represent video content for retrieval and repurposing, but to help computers understand textual stories. Schank and his students were deeply engaged in computational story understanding and story generation. The idea of story as a central organizing principle of human cognition is one of Schank's major contributions to artificial intelligence research. The challenge for video representation is to create a set of primitives which are pre-narrative and which enable a variety of video sequences to be constructed which support a multitude of possible narrative functions. Unlike the trans-linguistic aspirations of conceptual dependency, in Media Streams we do not try to reduce all video actions to media-independent primitives, but attempt to construct a vocabulary of primitives and rules of combination which articulate a semantics of action that is conditioned by the properties of the medium.

9.1.2.2. Dynamic Memory

For Schank, a dynamic memory is “one that can change its own organization when new experiences demand it.” (Schank 1982: 2). Schank’s work on dynamic memory emphasized the centrality of memory in intelligence and understanding. It provided a model for human and computational memory in which memory structures function as processing structures. The idea of a dynamic memory foregrounded reminding, indexing, and retrieval as basic cognitive processes which computers should seek to emulate.

Schank’s work in dynamic memory (Schank 1982) is an important precursor to Haase’s work in memory-based representation (Haase 1991; Haase 1993). An idea central to both is that understanding is the process by which we match new examples to previous ones indexed in memory and that memory structures themselves function as a representational and retrieval mechanism:

Finding the *right* one (that is, the one that is most specific to the experience at hand) is what we mean by understanding. Does this mean then that episodic memory structures and processing structures are the same thing? The answer is yes. It follows then that there is no permanent (i.e., unchangeable) data structure in memory that exists solely for processing purposes. Scripts, plans, goals, and any other structures that are of use in understanding must be useful as organizing storage devices for memories. These structures exist to help us make sense of what we have seen and will see. Thus, memory structures for storage and processing structures for analysis of inputs are exactly the same structures. (Schank 1982: 25)

The important difference between Haase’s work in memory-based representation and Schank’s approach in a dynamic memory is that Schank’s memory structures rely on abstract scripts or goals to organize and index concrete examples. In Haase’s memory-based representation, there are no categories or scripts under which cases are indexed. This is not to imply that Haase’s representation lacks organizing structures. It is in fact highly structure dependent. Functional similarity is determined by comparing concrete structures and indexing exceptions to structural similarity in memory. The differences from Schank’s dynamic memory are twofold: in Haase’s Mnemosyne, any example can emerge as a prototype due to the structure of the memory; and organizing structures and descriptive structures are the same. Any example can function as an organizing structure (prototype) for other examples in the memory—but

there cannot be no organizing examples. The structure of prototypes and spinoffs in memory are the representation. If one asks, where is the knowledge in a memory-based representation, the answer is that it emerges out of the interconnected differences and similarities of actual concrete bits of structure which have been parsed or segmented into coherent units and which point to one another through prototype and spinoff relations.

Our work in Media Streams makes use of Schank's and Haase's work on dynamic memories. Like Schank and unlike Haase, we seed our dynamic memory with a fairly detailed ontology and represent its contents using a semantic representation. Unlike Schank and like Haase, the exceptions to this ontology are indexed not by means of abstract scripts or goals, but by means of relations between prototypical examples.

9.1.2.3. Case-Based Reasoning

Case-based reasoning developed out of attempts to computationally implement Schank's ideas about dynamic memory (Riesbeck and Schank 1989; Seifert and others 1989). Case-based reasoning "solves new problems by adapting solutions that were used to solve old problems" (Riesbeck and Schank 1989: 25). Unlike most case-based reasoning systems, Media Streams is not itself a problem-solving application. However, video sequence retrieval can be thought of itself as a case-based reasoning problem: sequence retrieval-by-composition requires the retrieval and adaptation (through segmentation and resequencing) of video "cases." However, case-based reasoning has not been used in this way by Schank and his students. Rather than being represented, retrieved, and repurposed by its contents, video has been used as an accompanying data type within a case library of answers to questions or stories to tell in an instructional dialogue (Burke and Kass 1994; Osgood 1994).

For the past few years, Schank and his students have been conducting a large scale project to develop video databases for interactive corporate training applications. In this work, video is represented as if it were just textual dialogue that is represented in terms of the ideas it contains or the questions it answers. The video data is treated as if it were fully transparent and one need only represent the ideas behind it in order to fully represent its content. Schank does concede that this approach is designed for the needs of his current project and that it may prove inadequate for representing video that will be resegmented and/or repurposed (Schank 1993). This approach presents serious problems for a truly case-based video representation: representation and indexing must articulate the difference between sequence-dependent and sequence-independent aspects of video content, and then use this distinction to support the

retrieval and potential reindexing of cases when video segments are resequenced.

9.1.3. Lenat and Guha: Common Sense Knowledge Representation

The goal of the CYC project, which began in 1984, is to overcome the brittleness and domain specificity of all previous attempts at representing our common-sense knowledge about the world (Guha and Lenat 1994; Lenat and Guha 1990). Unlike a dynamic memory, CYC uses a variety of mostly rule-based inference mechanisms built on top of a very large, fixed ontology. It does not build up representations of the world through examples nor does it extend its semantic memory through connection to an episodic memory.

Since it is predominantly a semantic ontological structure with inference rules, CYC early on had a central problem in that it demanded that all its representations be consistent, correct, and non-contradictory. But as is well known from our own experience, human representational systems often represent things in multiple and contradictory ways. Marvin Minsky argues that it is the hallmark of intelligent behavior and activity to be able to manage contradictory representations (Minsky 1974; Minsky 1987). In order to attempt to manage contradictory representations, "microtheories" which translate axioms from one context to another were added to CYC in early 1990. Microtheories adhere to logical consistency; there can be blatant contradictions across microtheories, but each microtheory is expected to be internally consistent. This approach however does not deal with the loss of information that translation between representations necessarily implies. The *structure* of a representation (as in a dynamic memory or a frame-based system) contains information that a logical formalism cannot express. Translating a representation into a formalism in order to compare it to another representation translated into the same formalism implies a loss of information since the translation into the common formalism captures some features of each representation, but by necessity not all of them. The difference between logical formalist approaches to representation and memory-based approaches lies at the heart of a methodological debate in knowledge representation. In their work on KRL, Winograd and Bobrow articulate the distinction:

In designing KRL we have emphasized the importance of describing an entity by comparing it to another entity described in the memory. The object being used as a basis for comparison (which we call the *prototype*) provides a *perspective* from which to view the object being described. The details of the comparison can be

thought of as a further specification of the prototype. Viewed very abstractly, this is a commitment to a *wholistic* as opposed to a *reductionistic* view of representation. It is quite possible (and we believe natural) for an object to be represented in a knowledge system only through a set of such comparisons. There would be no simple sense in which the system contained a “definition” of the object, or a complete description in terms of its structure. However if the set of comparisons is large and varied enough, the system could find the answer to any question about the object that was relevant to the reasoning processes. This represents a fundamental difference in spirit between the KRL notion of representation, and standard logical representations based on formulas built out of primitive predicates. (Bobrow and Winograd 1985: 266).

Others argued that the content of a frame-based representation could be expressed entirely within a first-order predicate calculus representation (Hayes 1985). Winograd and Bobrow’s work grew out of attempts to resolve this debate (Winograd 1985), but in the intervening years the ranks have largely parted. CYC adheres to the belief that first-order logical formalisms can capture and translate the salient aspects of different representations. My contention is that this formalism is inadequate to the task of representing and managing incommensurability of representations and representation structures.

Other approaches, while sacrificing the completeness and correctness of logic, attempt to manage incommensurable representations in ways which do not suffer from problems of the loss of information in the translation between representational structures. Haase’s memory-based work links contradictory representations through indexing variant examples under prototypical examples. Case-based approaches manage contradictory representations through indexing of case exceptions and solutions for case adaptation. Agent-based approaches offer managers that link various (and often conflicting) agents into functioning dynamic systems. Some recent work has attempted to combine agent-based and case-based approaches by enabling a manager agent to have access to a case-library of solutions for arbitrating between conflicting agents (Travers and Davis 1993).

Regardless of its problems in dealing with contradictory representations, CYC has made considerable progress in creating representations of the everyday world. Most other research in knowledge representation simply ignores the need for a large common-sense knowledge base and thereby severely restricts its flexibility and scalability (Brachman and Levesque 1985). Because of the sheer size and thoughtfulness of the CYC project, it does provide useful insights into how to think about creating representations of objects, people, actions, time, and events.

Recently the CYC project has begun to apply its large semantic knowledge base and inference mechanisms to the representation and retrieval of still images and video (Lenat and Guha 1994). Surprisingly, these attempts fall prey to exactly the same criticism which Lenat himself levied against efforts to represent the physical world by natural language. Lenat argued that natural language was an inadequate representational system for representing knowledge about the world because it is not a designed representation (Lenat and Guha 1990). In other words, natural language is not designed in such a way as to capture the salient features of the world which are amenable to computational representation. Nevertheless, the CYC project makes a methodological error in its efforts to represent stills and video: it applies its representation language (a representation of the world) to video without redesigning it for the representation of video. What Media Streams does in contrast is create a representation language for video, in other words, a representation of a representation of the world.

CYC's efforts at video representation also suffer from information-theoretic assumptions common to most of artificial intelligence: the sender-receiver model of communication. In CYC, video sequences are represented as "information bearing objects with propositional content." CYC's semantic model is to capture what an image or a video *conveys* as opposed to what it *depicts* (Lenat 1994b). This model sidesteps the semantic issues raised by the Kuleshov Effect and the viewer's inferential activity. Lenat and Guha admit that their current approach may break down due to the particular sequence-dependent and sequence-independent semantics of video data (Guha 1994; Lenat 1993). With video, resequencing may change the given "propositional content" of any "information bearing object." They also admit that the particular semantic and syntactic structures of video sequences demand that a representational system have a way of dealing with the inferential consequences of what is depicted. It is in the role of supplying missing information that CYC may hold the most promise for video representation. By means of its large knowledge base, CYC—with a sufficiently revamped representation designed for video—could greatly enhance depiction oriented representation in the following ways:

- by enriching the semantic representation for retrieval beyond Media Streams' base ontology
- by providing educated guesses as to what may have been depicted but was not mentioned within a shot (since annotation is a necessarily incomplete task)
- by supplying inferences about what may have transpired between shots (greatly aiding in the creation of sequence retrieval systems which could understand what shots may be substituted for one another)

While CYC is an important and large scale project in knowledge representation, Media Streams offers the necessary representational and retrieval strategies which would enable its knowledge base to be used for video representation. Conversely, CYC's large common-sense knowledge base and inference rules could amplify annotation effort in Media Streams by supplying greater breadth and precision for its semantic and contextual representations.

9.1.4. Spatio-Temporal Logics

In addition to our own work, there have been some recent attempts to apply Allen's temporal logic (Allen 1985) to video sequence description and retrieval by extending his temporal logic through the addition of various spatial logics: the logic of objects moving in spaces over time; and the logic of cameras moving around fairly static scenes over time.

9.1.4.1. Spatio-Temporal Indexing

Researchers in Italy working within the visual languages community have extended temporal logic through a scheme for spatio-temporal indexing (Campanai and others 1992; Del Bimbo and others 1992; Del Bimbo and others 1993). In this work additional spatial operators are added to Allen's thirteen temporal operators.

The spatio-temporal indexing is used in a video retrieval system for footage of automobile motions at traffic intersections. The system uses an interactive 3-D model of a scene as a query formation interface. The representation and interface are well suited to the specific domain of auto intersection videos as well as to domains in which it is feasible and appropriate to create a full 4-D model of objects and events occurring in

one contiguous location. However, for video sequences which concatenate footage of discontinuous locations and discontinuous times (most cinematically structured video), this approach has two drawbacks for video indexing and retrieval:

- the spatial coordinates are object-centered in a real 3-D space as opposed to screen centered in a non-real cinematic space
- the notion of “video sequence” in the representation is of a series of events within a scene as opposed to a series of shots in a sequence (i.e., the notion of sequence is non-cinematic)

Del Bimbo’s work uses object-centered coordinate systems such that video shots of the same scene from different viewpoints share the same descriptions (Del Bimbo and others 1993: 90). This work raises interesting questions about the relationship between coordinate system representations used in computer graphics and the coordinate systems of cinematic spaces. Del Bimbo’s approach has clear advantages for contiguous real world spaces, but falls short for cinematically constructed spaces with non-real-world geometries. For example, real-world 3-D spatial representations have trouble representing the geometry of houses in which doors alternately lead to different rooms as in the *Nightmare on Elm Street* movies. Even in more traditional cinema it is often the case that the location of exits and entrances to a space will be governed by screen compositional and narrative principles, rather than by architectural ones. In cinematic video sequences the coordinate system of the screen is as important as the coordinate system of objects since screen positions and geometries can create the impression of new object positions and geometries which if projected into a 3-D coordinate system would not be consistent. It remains to be seen how Del Bimbo’s object-centered coordinate system would work for more cinematic video.

In this work on video sequence retrieval, the sequences are understood to be merely one continuous shot of a series of events within one contiguous location, like two cars driving by each other at an intersection. But in film-theoretic understanding of sequences, a sequence is composed of two or more shots. The sequence always involves a transition between shots usually involving a temporal discontinuity and/or a spatial discontinuity. Therefore, Del Bimbo’s video “sequence” retrieval is really a form of retrieval of shots of complex action sequences. Creating a video sequence retrieval system involves representing shot transitions. Understanding the function of transitions in the construction of video sequences is an essential

part of creating a representation of video that facilitates the reusability of video shots in the retrieval and creation of video sequences.

9.1.4.2. Settings

Some research relevant to issues of video representation came to my attention in the final days of writing this thesis. Alan Parkes at the University of Lancaster in England conducted research in the late 1980's on representing the content of videodisc materials for interactive intelligent tutoring systems (Parkes 1989b). Parkes developed a notion of a basic unit of representation of videodisc content that he called a "setting." Parkes defines a setting in the following way:

The *setting* is essentially defined as being a group of one or more *still images* which display the same *objectively visible "scene"*. Effectively, the *objectively visible* dimension of an image consists of the *physical objects* represented in an image and the relationships between those physical objects which can be *seen*, and are not *inferred-to-be-present*, as it were. Still images have a *continuum of meanings* (Parkes 1989a) ranging from the *objectively visible* (what they objectively "show"), through *event-ambiguous*, where a member of a *set of events* could be assigned as being what the image could be inferred to be displaying a "moment" of, and finally to *event-determined*, where some other information beyond that contained in the actual image itself leads the viewer to infer that a *particular event* is represented (in "snapshot" form). The author's contention is that *descriptions actually applied to the image itself should be as objective as possible*, because it is only by doing this that the maximum flexibility of use of the arbitrary image can be guaranteed. (Parkes 1989b)

Although Parkes' uses of terminology like "objective" is philosophically overdetermined, his notions of description are very much in line with our own. His idea of a "continuum of meanings" for still images is analogous to our distinction between sequence-independent and sequence-dependent semantics. His emphasis on describing objects, relations, and camera motions so as to guarantee the "maximum flexibility of use of the arbitrary image" is very akin to the idea behind our physically-based description for repurposing video content. Parkes also makes an important point in his essay about the insufficiency of keywords for image description because they cannot express relationships between terms (Parkes attributes this insight to a personal communication with J. Gecsei). Parkes also brings up the important and often misunderstood issue of the problem of

identity in the description of image contents. He argues for a rigorous distinction between descriptions of objects of the “same type” and descriptions of the “same actual object.” This distinction is the difference between describing two objects as belonging to the same class or as being the same instance of the same class. The distinction though is more problematic than Parkes acknowledges since rules for identity in video are played with in ways not seen in the physical world or in still images. The clarity of the distinction is blurred when identity can be constructed across sequences of images. The notion of “same actual object” is a problematic one in video. Nevertheless, the issue is an important one to raise and to address. In our own work we are still exploring the space of this problem for video.

In Parkes’ scheme, transitions between still images are conceptualized as relations between settings. These relations involve pan, tilt, roll (cantiing), and zoom operations of an imaginary camera, while changes in image contents are described as modifications of settings. Parkes approach has the keen insight of conceiving of a videodisc as a series of still images which can be shown in any sequence. Thus he represents video stills and still images of a laserdisc as *frames* in a large space of possible video sequences. He develops a logic of transitions between settings that charts paths through that space. His approach runs in to trouble in trying to cope with multiple overlapping camera motions and modifications of settings contents. His camera representation conflates trucking shots with pans and zooms so is somewhat cinematographically inaccurate. He, like Del Bimbo, also assumes that video represents an actual coherent physical 3-D space rather than facilitates the construction of cinematic spaces which may have irreal and physically inconsistent geometries. Consequently, he does not offer a representation of discontiguous and discontinuous transitions between settings. He also does not provide a framework for the semantic representation of image contents nor a sufficient representation of action and motion within the frame—though as mentioned above Parkes understands some of the essential parameters such a representation must satisfy. In his as yet unpublished recent work with his students at Lancaster, it appears that Parkes is attempting to address the limitations of his earlier research.

9.2. Integrative Approaches to Video Representation

Work in knowledge representation and film theory has laid the foundation for knowledge representation for video. However, because of the need for human annotation, building workable systems for representing video for retrieval and repurposing requires the integration of signal-based parsing and human-computer interface technologies as well. Media Streams is such a system. Over the past three years a few other researchers have attempted to design and build systems which incorporate elements of knowledge representation, film theory, signal-based parsing, and human-computer interface to varying degrees. These integrative research projects attempt to solve the entire problem of video representation for retrieval and repurposing. They necessarily incorporate representations and interfaces to support preprocessing, annotation, browsing, retrieval, and repurposing (or at least resequencing) of video content.

The MIT Media Laboratory has been a hotbed of activity in this area and has produced some of the best work in the field. The other site at which a large scale integrative effort has been underway is the Institute for System Sciences in Singapore.

9.2.1. Institute for System Sciences

HongJiang Zhang and Stephen Smoliar have been building a series of tools and applications for parsing, visualizing, browsing, representing, and retrieving video (Smoliar and others 1994; Zhang and others 1993; Zhang and Smoliar 1994). Zhang and Smoliar have developed a multiple pass algorithm for automatically detecting fades in video that builds on the histogram techniques developed by Nagasaka and Tanaka (Nagasaka and Tanaka 1992). As in our own research, Zhang and Smoliar have contributed their own innovations as well as integrated earlier efforts into a unified system. Like Media Streams, they use and have extended the XY-T representation for video data begun in systems by Ueda (Ueda and others 1991) and Elliott (Elliott 1993). Though not the first to demonstrate this idea, Zhang and Smoliar have developed a 3-D manipulation and sampling interface for XY-T representations that makes them a useful interactive browsing tool.

Building on their work in signal-based parsing and interface techniques, Zhang and Smoliar have recently begun to design a system for representing the semantic content of video data (Smoliar and others 1994). Using a frame-based structure for knowledge representation (Bobrow and Winograd 1985; Haase 1994; Minsky 1974; Winograd 1985), they build up domain-specific semantic topic hierarchies for representing video

content. Their approach is clip-based, domain-specific, and does not have a controlled vocabulary, hence, its applicability to creating large archives of reusable video appears limited. However, within specific domains, their frame-based approach should prove vastly superior to contemporary keyword-based approaches as was evidenced in the case of similar work done by researchers at Coopers and Lybrand in their Semantic Refinery system developed for Monitor Television (Clippinger and others 1992; Clippinger and others 1993). Zhang and Smoliar have focused on the domain of news footage and (unlike the Semantic Refinery system) have developed specialized parsers which combine signal-based and semantic techniques for segmenting news broadcasts (Zhang and others 1994). Their parsing model of the content of new broadcasts builds on earlier work by Swanberg (Swanberg and others 1993). It remains to be seen how they will use their promising preliminary work on video segmentation and manipulation to solve the problems of video retrieval and resequencing.

9.2.2. MIT Media Laboratory

Work on video representation has been a part of the Media Laboratory since its inception. In recent years, the focus has decisively shifted from research on pixel and bit level manipulation of video to representations for manipulating video by its content (Levitt and Davenport 1987). In Nicholas Negroponte's words, the Media Lab is inventing the "bits about bits" that all digital media will eventually have as accompanying descriptions of their contents. In various research groups at the Media Laboratory content-oriented work on video has focused on different levels of representation. In the Garden (structured video) and the Vision and Modeling Groups (things and stuff) work on low-level representations of video content has until recently existed largely independently of high level work on story level representations developed in the Interactive Cinema Group. Attempts to investigate the mid-level representations of video content which might inhabit the space between, and possibly connect, low-level (pixels and objects) and high-level representations (stories) are a more recent development.

There was some early research in representing video for resequencing (Rubin 1989; Rubin and Davenport 1989) that built on top of Bloch's work on video representation and sequencing (Bloch 1987). Rubin's work came close to conceptualizing a stream-based representation. It had a notion of action descriptions which had temporal relations among themselves, however, they were restricted to individual shots and could not extend across shot boundaries. This work also raised issues of a content rich production pipeline, but did not fully address problems of annotation for large video archives.

In addition to our own work, recent projects in the Interactive Cinema Group and a project sponsored by BT (British Telecom) connecting that group, the Vision and Modeling Group, and the Machine Understanding Group in which my research has been conducted, have attempted to research and develop integrative systems for preprocessing, annotating, browsing, retrieving, and repurposing video content.

9.2.2.1. Cinematic Primitives for Multimedia and Stratagraph

The Interactive Cinema Group headed by Glorianna Davenport has been addressing the concerns of filmmakers working in a new digital medium. Early experiments in “elastic cinema” have lead the group to attempt to solve problems of cinematic content representation (Davenport 1987; Mackay and Davenport 1989). Davenport and her students worked on isolating basic components of the cinematic recording and production process with an eye toward identifying which aspects can be computationally represented (Davenport and others 1991). In this work and in work based on it, Thomas Aguierre-Smith applied ideas taken from anthropological descriptive practice to the task of representing video in a reusable archive that can be accessed and annotated by multiple users. The stream-based representation in Media Streams arose out of early discussions between Thomas Aguierre-Smith and myself about video representation. His system, *Stratagraph*, supports stream-based textual annotation and retrieval of video from a laserdisc archive (Aguierre-Smith 1992; Aguierre-Smith and Pincever 1991). Although *Stratagraph* is an important comparative milestone in stream-based video representation, it does not offer a semantic representation of video content and is keyword-based. Therefore it is limited in its ability to support video annotation for large scale content-based retrieval and repurposing.

Two other projects at MIT have improved on *Stratagraph* by adding additional representational structures to its underlying framework.

9.2.2.1.1. Homer and Sequencer

Lee Morgenroth, under the direction of Glorianna Davenport, has built video story generation and editing tools on top of Aguierre-Smith’s *Stratagraph* system. His story generation program is called *Homer* (Morgenroth 1992) and his editing application connected to *Homer* and *Stratagraph* is called *Sequencer* (Davenport and Morgenroth 1994). *Homer* has an interface of nested hierarchical story blocks which a user can manipulate to author a story query. The results of this story query can be refined, trimmed, and resequenced in *Sequencer*. *Homer* adds additional

sequence specific representations about which video segments might fit well together which are written and stored in FRAMER (Haase 1994; Haase and Sack 1993). Homer provides an intuitive model of and interface to the story construction process. However, Homer still relies largely on the keyword-based descriptive model of Stratagraph. What is revealing is that in practice Morgenroth has by default introduced semantic categorization schemes like objects, locations, characters, etc. into Stratagraph. In describing video for repurposing, the need for a minimal ontology has made itself apparent. However, these descriptions in Homer/Stratagraph are not structured categories with a controlled generative vocabulary, but a user-specific practice of description that does not ensure a consistent reusable representation of video content for multiple users.

9.2.2.1.1. Algebraic Video

At the MIT Laboratory for Computer Science's Programming Systems Research Group, Ron Weiss has developed an extension to Stratagraph's model of stream-based representation that he calls "nested stratification" (Weiss and others 1994). In nested stratification, descriptions which temporally contain other descriptions are related to them hierarchically in the representation such that browsing and playout can make use of these structures of temporal encapsulation. Weiss's *Algebraic Video System* uses nested stratification to create a series of basic compositional operations (such as union, intersection, etc.) between video descriptions to structure retrieval and authoring. The Algebraic Video System uses non-semantic and unrestricted text-based descriptors and as such suffers from same scaling and consistency problems of other similar approaches. Furthermore, in its current incarnation, the system does not address the problem of how nested stratifications are written. Although it does employ many useful automatic parsing techniques, the need for an annotation system is ignored and mistakenly considered to be a problem which purely automatic methods will solve.

9.2.2.2. BT Project

The BT Project at the MIT Media Laboratory is attempting to integrate technologies operating at various levels of video representation into a combined system for accessing the contents of video databases (Pentland and others 1993b; Pentland and others 1994a). Work in the Vision and Modeling Group on semantics-preserving image compression attempts to automatically extract a small set of salient image features which preserve structural relations between image components and therefore aid in

content-based compression and retrieval (Pentland and others 1994b). Pentland and his students have developed an application using this technology called *Photobook* that also makes use of FRAMER to store and compare textual descriptions of images in order to reduce the space of images to be searched using features extracted by semantics-preserving compression methods (Haase 1994; Haase and Sack 1993). Recently, Morgenroth and Pentland have integrated Photobook into the Sequencer application enabling the interleaving of textual and image-based retrieval of video shots. This research may lead to the development of human-in-the-loop annotation and retrieval methods described at the end of Chapter Four.

9.2.2.3. Domain-Specific Video Resequencing Systems

Media Streams attempts to provide a representational framework to support video retrieval and repurposing across content domains. In addition to Homer, other research prototypes for *domain-specific* video resequencing have been developed over the past few years by researchers at the Media Laboratory. These prototypes have each contributed useful ideas to the problem of video story generation which may inform the creation of video sequence generation systems which can work across content domains.

9.2.2.3.1. Suburban Ontology: Electronic Scrapbook

The *Electronic Scrapbook* provides an interface and a representational framework which enable home video users to annotate, retrieve, and resequence video clips (Bruckman 1991). Amy Bruckman developed the Electronic Scrapbook within the Interactive Cinema Group to address the problem of the hours of unwatched and unedited video that home users create. Bruckman developed a “suburban ontology” designed to represent the world of upper middle class, suburban home video users (the videos in her system were of her own family). The suburban ontology developed in the Electronic Scrapbook used an early knowledge representation language written by Prof. Kenneth Haase called ARLOtje that has since been reimplemented on top of FRAMER. Bruckman’s work offers two important insights for video generation systems: an ontology tailored to the home video domain could aid in annotation and sequence generation within that domain; and simple domain specific story models (e.g., shots of a child at different ages or shots of two children at significant events in their lives) could use this ontology to make sequences relevant to users in the domain.

9.2.2.3.2. Montage Rules and Ideology: Splicer

Investigating the role of ideology in the news and its possible use as a principle of construction for repurposing video content, Warren Sack and Abbe Don built a prototype video resequencing system called *Splicer* (Sack and Don 1993). *Splicer* uses a clip-based, dialogue-centric video representation that describes the content of video in terms of the relationships between actors and their roles in an ideological system. This actor-role representation is used by Sack in later work to analyze and retrieve news stories (Sack 1994a) and constitutes a theory of how to represent point of view and bias in the news (Sack 1994b). In *Splicer*, these actor-role representations are used by various rhetorical strategies (e.g., point-counterpoint) to structure video sequences. Don's work on narrative construction and point-of-view (Don 1990; Laurel and others 1990; Oren and others 1990) also informs the design of this system through the representation of who is speaking and who is being spoken to (which can be individuals or groups). The key contribution of the *Splicer* system is the representation of video within a framework of ideological stances, affiliations, and exchanges that enables the repurposing of video into micro-documentaries expressing distinct points of view.

9.2.2.3.3. GPS and Montage: IDIC

In the fall of 1992, Warren Sack and I began a critical evaluation through reimplementing of AI technologies for video sequence generation. We began with the design and construction of a prototype system, called *IDIC*, which uses Newell and Simon's General Problem Solver to construct video sequences (Newell and others 1963). We sought to prove that GPS would not work for video story generation but were chagrined by our system's ability to use GPS to make some interesting sequences (Sack and Davis 1994). *IDIC* uses GPS operators to construct a story plan that retrieves video annotated in terms of basic narrative units (e.g., fight, rescue, threaten, negotiate). We discovered that a GPS operator with its pre-conditions, post-conditions, and the action that links them functions very much like a shot transition in a video sequence. A video sequence can be constructed by concatenating shots in which the action linking two shots is inferred but not shown (as in the classic Kuleshov example). A montage of shots satisfying the pre-conditions and post-conditions of a series of chained GPS operators resulted in a sequence of inferable actions. We also found that the regular planning criteria of looking for the shortest plan to solve a goal was not necessarily the criteria that one wanted for generating interesting trailers. At first, GPS kept churning out overly-short sequences, until we replaced the sort function's predicate with ">" rather than "<". The use of the ">" predicate allowed *IDIC* to pick the longest

plan (i.e., to look for the next step with the most unfulfilled preconditions rather than the one with the least unfulfilled preconditions) and hence produce longer, more interesting sequences. Using our modified version of GPS, IDIC was able to generate new, short *Star Trek: The Next Generation* trailers (albeit silent ones) out of parts of existing ones.

In addition to the insight about the analogy between GPS operators and shot transitions, we also found that the choice of materials in our system—the specific visual properties of the shots we annotated—greatly enhanced the efficacy of the sequences the system could generate with these reusable parts. The power of visually based stories using well chosen parts is also seen in Rachel Strickland’s narrative construction toolkit, *Backyard Transformations*, which supported manual sequence construction with a rich set of manipulatives (Strickland and Wright 1990). The content representation in Media Streams is designed to enhance the combinatorial richness of video content and to facilitate the automatic selection of video segments which satisfy such criteria. Finally, in IDIC we encountered the importance of playback speed for the intelligibility of action in silent visual sequences (we found that the trailers generated by IDIC needed to be played back at 50% normal speed when the narration was turned off). What IDIC reiterated for us was the importance of providing minimal narrative cues to support the viewer’s inferential activity, which, if leveraged appropriately, is the most important processing element of any automatic video generation system.

9.3. Comparison To Related Work

The systems we have discussed in this chapter have contributed in various ways to the problem domain our work attempts to address. In their successes and their limitations they help to articulate the space of solutions researchers are developing to represent video content for retrieval and repurposing. Media Streams is the only system that attempts to solve the problems of reading and writing content annotations with an iconic visual language. Most importantly, it is also the only system that combines all of the elements we believe are crucial to solving the problem of creating reusable media. We can compare Media Streams to the systems we have discussed in the following table:

Table 8.

Practitioners	Stream Based	Semantic Representation	Controlled Vocabulary	Signal Based Parsing	Video Semantics	Video Syntax
Davis	●	●	●	●	●	●
Del Bimbo	●	●	●			
Sack/Davis	●	●	●			
Morgenroth	●	●		●		
BT Project	●			●		
Weiss/Duda	●			●		
Aguierre-Smith	●					
Bloch		●	●		●	●
Rubin		●	●		●	●
Parkes		●	●		●	
Lenat/Guha		●	●			
Schank et al		●	●			
Sack/Don		●	●			
Bruckman		●	●			
Smoliar/Zhang		●		●		

Roughly half of the systems we have considered use some form of stream-based representation. Many of them employ various types of semantic representation. If they do not also use a controlled vocabulary, they invite unstructured growth of their semantic hierarchies that usually leads to the creation of incommensurable representations. Fewer systems combine semantic representation with signal-based parsing. Although they do not use any forms of signal-based parsing, only the systems of Bloch and Rubin also attempt to articulate a representation of both video semantics and video syntax. Only Media Streams exhibits all of the above features and combines them in a system designed to enhance the repurposability of video content across domains. Unlike any of the other systems mentioned, Media Steams enables humans and computers to work together to create sharable and durable representations of video content which support the retrieval-by-composition of new video sequences out of parts of existing ones.

We assert that the features used to differentiate the various systems in the above table are necessary for the design and implementation of systems for representing video content for retrieval and repurposing. The question which future research and development will answer is whether these features are also sufficient.



Chapter Ten

Conclusions and Future Directions

10. Conclusions and Future Directions

10.1. Conclusions

In this thesis I set out to solve the problem of representing video content for retrieval and repurposing. After clarifying the need for a language for representing video content, I explained the techne-centered methodology on which my research has been based. I explored the present and future domains of use for my technology—stock footage houses and Garage Cinema makers—and examined their changing dynamics of production, cost, and use. The central contributions of my research were articulated in a discussion of the issues involved in representing video for retrieval and repurposing and of the technologies I developed for writing and reading these representations. I described the results of the User Study conducted to validate the claims made in my research that my system is learnable, that users can reuse each others' descriptive effort, and that different users' annotations of the same footage are semantically convergent. Finally, I compared my work to other relevant work in knowledge representation and video manipulation systems.

I have designed, implemented, and evaluated a representation language for video content that enables humans and machines to work together to annotate, browse, retrieve, and repurpose video in ways which were not possible before. My system, Media Streams, is a stream-based iconic visual language that makes use of the semantic and syntactic properties of video in order to enable retrieval-by-composition of video sequences from an archive of annotated footage.

Media Streams' representation and retrieval technologies employ cinematic concepts of space, time, character, and action. Its representational framework uses a hierarchically structured semantic vocabulary of composable primitives which overcome the limitations of keyword-based systems. Media Streams' iconic visual language overcomes the limitations of text-based approaches to reading and writing representations of video content. On Media Time Lines, semantic descriptors are used to articulate the episodic temporal structures of video content. Through indexing of these episodic relationships between descriptions in the semantic hierarchy of descriptors, Media Streams articulates the differences between the sequence-dependent and sequence-independent semantics of video data. Finally, Media Streams' representations, algorithms, and interfaces enable users to annotate and create-by-retrieving video sequences which repurpose video content.

Important next steps within the research agenda of Media Streams include:

- the development of more complete representations for audio content
- the incorporation of textual data such as dialogue and closed-captioning in the representation
- the structural integration of Media Streams' mid-level content representation to additional low-level representations generated by signal-based parsing and to high-level representations of narrative structure and function

My research provides a technological and theoretical foundation for the exploration of questions of video representation, retrieval, and repurposing. This intellectual and social endeavor is just beginning. It is my sincere hope that my research may initiate a process of investigating more questions than it itself has answered.

10.2. Future Directions

10.2.1. Artificial Intelligence, Film Theory, and Media Technology

We live in interesting times. Though ancient Chinese wisdom saw this as a curse, I see it as an opportunity. Disciplinary boundaries which have evolved over the past few centuries between the sciences, humanities, and arts are breaking down under the pressures of new forms of technological and theoretical practice. Work in media technology holds the promise of marshaling the intellectual resources of diverse disciplines to solve problems which cannot be addressed within the confines of any one discipline. Representing video for retrieval and repurposing is precisely such a problem. Representing video content is an intellectually challenging, economically astute, and socially relevant research agenda for the ever self-reinventing discipline of artificial intelligence. For film theorists, new media technologies enable the tools of analysis to be transformed into tools of construction. Technologically coupled forms of interpretive and (de)constructive practice can transform cultural materials once preserved, analyzed, and revered within the academy into tools of educational renaissance and cultural change. The task at hand is to create common ground between artificial intelligence and film theory so that practitioners within and between these disciplines can work together to

solve problems which exceed their respective methodological and technological grasp. My work is both a gauntlet thrown down to begin that ensuing challenge and an open hand reaching across the disciplinary divide separating technologists and humanists.

10.2.2. Towards Garage Cinema

What this research is really about is getting my hand inside that television set I loved, worshipped, and grew up with as a child. It is about the Promethean act of taking the fire from the box and setting the world alight. If I were a young teenager in 1998, I would want to play with Media Streams (or its descendants) to make movies from video I found on the Net, recorded from television, and shot with my camcorder. Maybe you do too. It is my contention that you, or your children, will.

What especially gives me confidence in this vision of the future is the 3 1/2 hours I recently spent with Henry Jenkins III and his 13 year-old son Henry Jenkins IV using Media Streams. For Henry IV this experience was part of his day at “school” (Henry IV is participating in home schooling this year). I talked with Henry IV about my project, its goals, and showed him how to use the system. In talking about how one might want to represent video for retrieval and repurposing, Henry IV derived most of Media Streams’ categories from first principles. He enjoyed figuring out and using the icons. He was adept at the system in 30 minutes, making sequences through retrieval-by-composition methods in about an hour, and after 3 1/2 hours when I had to leave wanted to know how soon he could come back. He is thirteen. The future is his—I hope I can play in it with him and make some interesting tools/toys for him and his friends.

Henry IV made many sequences with Media Streams’ retrieval-by-composition technology. One sequence he wanted to make especially sticks in my mind because of the playfulness it assumes so effortlessly and the ways it encourages me to think and push the technology: a crowd of Rastafarian centaurs walking over a bridge in North America during the Elvis period. If Media Streams technology were to be adopted as a standard for video annotation and retrieval, and if video extraction and compositing technology advances and becomes affordable to consumers, Henry IV may be able to make that movie at his house in a few years.

Just as we often find it hard to imagine our own civilization before the advent of widespread literacy in the 17th and 18th centuries, in the next century our descendants will find it hard to understand that while everyone watched movies, videos, and TV, so few had the tools to make them. The vision of Garage Cinema attempts to convey the radical changes in practices of production, distribution and use that video representation

technology, like Media Streams, will make possible. It may be hard to conceptualize a world in which you engage in a daily practice of making movies from parts of existing ones to communicate and play with others—your grandchildren will not understand how you ever lived without it. Watching what teenagers are already doing with the primitive tools of camcorders and computers today is inspiring. Imagining what they would be able to do within a global media archive indexed with Media Streams initiates the journey toward the other side of a paradigm shift in media technology and human communication that we are about to begin.

I'll see you in the garage.



References

References

- Abelson, Harold, Gerald Jay Sussman, and Julie Sussman. Structure and Interpretation of Computer Programs. The MIT Electrical Engineering and Computer Series, Cambridge, Massachusetts: The MIT Press, 1985.
- Ackermann, Edith. "The Agency Model of Transactions: Toward an Understanding of Children's Theory of Control." In Constructivism, ed. Idit Harel and Seymour Papert. 376-379. Norwood, New Jersey: Ablex Publishing Company, 1991.
- Aguiere-Smith, Thomas G. "If you could see what I mean... Descriptions of Video in an Anthropologist's Video Notebook." Master of Science Thesis, Massachusetts Institute of Technology, 1992.
- Aguiere-Smith, Thomas G. and Natalio Pinciver. "Parsing Movies in Context." In: Proceedings of Summer 1991 USENIX Conference in Nashville, Tennessee, Usenix Association, 157-168, 1991.
- Akutsu, Akihito, Yoshinobu Tonomura, Yuji Ohba, and H. Hashimoto. "Video Indexing Using Motion Vectors." In: Proceedings of SPIE Visual Communications and Image Processing '92 in Boston, Massachusetts, 343-350, 1992.
- Allen, James F. "Maintaining Knowledge about Temporal Intervals." In Readings In Knowledge Representation, ed. Ronald J. Brachman and Hector J. Levesque. 510-521. San Mateo, California: Morgan Kaufmann Publishers, Inc., 1985.
- Apple Computer. Macintosh Common Lisp Reference. Cupertino, California: Apple Computer, 1993a.
- Apple Computer. QuickTime Developer's Guide. Cupertino, California: Apple Computer, 1993b.
- Apple Multimedia Lab. The Visual Almanac. San Francisco, California: Apple Computer, 1989.
- Archive Films. "Archive Films Demo Reel." New York: Archive Films, 1992.
- Arman, Farshid, A. Hsu, and M.-Y. Chiu. "Feature Management for Large Video Databases." In: Proceedings of SPIE Conference on Storage and Retrieval for Video and Image Databases in San Diego, California, 1992.

- Arndt, Timothy and Shi-Kuo Chang. "Image Sequence Compression by Iconic Indexing." In: Proceedings of 1989 IEEE Workshop on Visual Languages in Rome, Italy, IEEE Computer Society Press, 177-182, 1989.
- Arons, Barry. "Interactively Skimming Recorded Speech." Ph.D. Thesis, Massachusetts Institute of Technology, 1993a.
- Arons, Barry. "SpeechSkimmer: Interactively Skimming Recorded Speech." In: Proceedings of UIST'93 ACM Symposium on User Interface Software Technology in Atlanta, Georgia, ACM Press, 1993b.
- Baddeley, Alan D. "Memory Theory and Memory Therapy." In Clinical Management of Memory Problems, ed. Barbara A. Wilson and Nick Moffat. 5-27. Rockville, Maryland: Aspen Systems Corporation, 1984.
- Barthes, Roland. "The Sequences of Actions." In Patterns of Literary Style, ed. Joseph Strelka. University Park, Pennsylvania: State University of Pennsylvania Press, 1971.
- Barthes, Roland. Image Music Text. Translated by Stephen Heath. New York: Hill and Wang, 1977.
- Barthes, Roland. Mythologies. Translated by Annette Lavers. New York: Hill and Wang, 1972.
- Bates, Joseph. "AAAI-94 Invited Talk on 'The Synergy of Artificial Intelligence, Art, and Interactive Entertainment'." In: Proceedings of AAAI-94 in Seattle, Washington, 1994a.
- Bates, Joseph. "CFP: Papers on Arts and Entertainment for AAAI-94." Electronic Mail Message. 1994b.
- Bazin, André. What is Cinema? Vol. 1. Translated by Hugh Gray. Berkeley: University of California Press, 1971.
- Biddick, Kathy. Personal Communication. 1994.
- Bliss, Charles Kasiel. Semantography-Blissymbolics. 3rd ed., Sydney, N.S.W., Australia: Semantography-Blissymbolics Publications, 1978.
- Bloch, Gilles R. "From Concepts to Film Sequences." Internal Document. New Haven, Connecticut: Yale University Department of Computer Science, 1987.
- Bobick, Aaron F. "Representational Frames in Video Annotation." In: Proceedings of IEEE Signals and Systems Conference in Asilomar, California, 1993.

- Bobrow, Daniel G. and Terry Winograd. "An Overview of KRL: A Knowledge Representation Language." In Readings in Knowledge Representation, ed. Ronald J. Brachman and Hector J. Levesque. 263-286. San Mateo, California: Morgan Kaufmann Publishers, 1985.
- Bordwell, David. The Cinema of Eisenstein. Cambridge, Massachusetts: Harvard University Press, 1993.
- Bordwell, David and Kristin Thompson. Film Art: An Introduction. 3rd ed., New York: McGraw-Hill Publishing Company, 1990.
- Bordwell, David. Making Meaning: Inference and Rhetoric in the Interpretation of Cinema. Harvard Film Studies, Cambridge, Massachusetts: Harvard University Press, 1989.
- Bordwell, David. Narration in the Fiction Film. Madison: University of Wisconsin Press, 1985.
- Bordwell, David, Janet Staiger, and Kristin Thompson. The Classical Hollywood Cinema: Film Style and Mode of Production to 1960. New York: Columbia University Press, 1985.
- Brachman, Ronald J. and Hector J. Levesque, ed. Readings In Knowledge Representation. San Mateo, California: Morgan Kaufmann Publishers, Inc., 1985.
- Bruckman, Amy. "The Electronic Scrapbook: Towards an Intelligent Home-Video Editing System." M.S. Thesis, Massachusetts Institute of Technology, 1991.
- Burch, Noël. Theory of Film Practice. Translated by Helen R. Lane. Princeton: Princeton University Press, 1969.
- Burke, Robin and Alex Kass. "Refining the Universal Indexing Frame to Support Retrieval of Tutorial Stories." In: Proceedings of AAAI-94 Workshop on Indexing and Reuse in Multimedia Systems in Seattle, Washington, edited by Catherine Baudin, Marc Davis, Smadar Kedar, and Daniel M. Russell, AAAI Press, 1-11, 1994.
- Burrough, Bryan. "Polar Privation: Antarctic Life Proves Hard Even for Those Who Love Their Work." Wall Street Journal, December 10, 1985, Section 1, Page 1, Column 1.
- Burroughs, William. Nova Express. New York: Grove Press, 1964.
- Campanai, M., Alberto Del Bimbo, and P. Nesi. "Using 3D Spatial Relationships for Image Retrieval by Contents." In: Proceedings of 1992 IEEE Workshop on Visual Languages in Seattle, Washington, IEEE Computer Society Press, 184-190, 1992.

- CBS News Archives. "CBS News Archives Sales & Licensing Brochure." New York: CBS News Archives, 1992.
- Chakravarthy, Anil S. "Toward Semantic Retrieval of Picture and Video Clip Captions." Submitted to IEEE Computer Special Issue on Content-Based Picture Retrieval Systems, 1995.
- Chakravarthy, Anil S., Kenneth B. Haase, and Louis M. Weitzman. "A Uniform Memory-Based Representation for Visual Languages." In: Proceedings of European Conference on Artificial Intelligence in Vienna, Austria, 1992.
- Chang, Shi-Kuo. "Visual Languages and Iconic Languages." In Visual Languages, ed. Shi-Kuo Chang, Tadao Ichikawa, and Panos A. Ligomenides. 1-7. New York: Plenum Press, 1986.
- Chang, Shi-Kuo, G. Costagliola, S. Orefice, G. Polese, and B. R. Baker. "A Methodology for Iconic Language Design with Application to Augmentative Communication." In: Proceedings of 1992 IEEE Workshop on Visual Languages in Seattle, Washington, IEEE Computer Society Press, 110-116, 1992.
- Clippinger, John, Henrik Sandell, and Philp Werner. A Multimedia Refinery. Coopers & Lybrand Advanced Technology Group, 1992. Digital Media Project Report 4.
- Clippinger, John, Henrik Sandell, and Philp Werner. Video Catalogue. Coopers & Lybrand Advanced Technology Group, 1993. Digital Media Project Report 7.
- Connor, Bruce. "A Movie." USA: 1958.
- Cooper, Douglas. The Cubist Epoch. London: Phaidon Press Limited, 1970.
- Davenport, Glorianna. "New Orleans in Transition, 1983-1986: The Interactive Delivery of a Cinematic Case Study." In: Proceedings of The International Congress for Design Planning and Theory in Boston, Massachusetts, 1-7, 1987.
- Davenport, Glorianna and Lee H. Morgenroth. "Video Database Design: Convivial Storytelling Tools." Internal Document. Cambridge, Massachusetts: MIT Media Laboratory, 1994.
- Davenport, Glorianna, Thomas G. Aguiere-Smith, and Natalio Pincever. "Cinematic Primitives for Multimedia." IEEE Computer Graphics and Applications 11.4 (July 1991): 67-75.
- Davis, Marc. "Knowledge Representation for Video." In: Proceedings of AAAI-94 in Seattle, Washington, AAAI Press, 120-127, 1994a.

- Davis, Marc. "Media Streams: Representing Video for Retrieval and Repurposing." In: Video Proceedings of Second ACM International Conference on Multimedia in San Francisco, California, ACM Press, 1994b.
- Davis, Marc. "Media Streams: An Iconic Visual Language for Video Annotation." Teletronikk 4.93 (1993a): 59-71. (Also available on the WorldWideWeb at: http://www.nta.no/teletronikk/4.93.dir/Davis_M.html)
- Davis, Marc. "Media Streams: An Iconic Visual Language for Video Annotation." In: Proceedings of 1993 IEEE Symposium on Visual Languages in Bergen, Norway, IEEE Computer Society Press, 196-202, 1993b.
- Davis, Marc. "Director's Workshop: Semantic Video Logging with Intelligent Icons." In: Proceedings of AAAI-91 Workshop on Intelligent Multimedia Interfaces in Anaheim, California, edited by Mark Maybury, 122-132, 1991.
- Deardorff, E., T. D. C. Little, J. D. Marshall, D. Venkatesh, and R. Walzer. "Video Scene Decomposition with the Motion Picture Parser." In: Proceedings of IS&T/SPIE Symposium on Electronic Imaging Science and Technology, 1994.
- Degen, Leo, Richard Mander, and Gitta Salomon. "Working with Audio: Integrating Personal Tape Recorders and Desktop Computers." In: Proceedings of Human Factors in Computing Systems: CHI '92 in Monterey, California, edited by Penny Bauersfeld, John Bennett, and Gene Lynch, ACM Press, 413-418, 1992.
- Del Bimbo, Alberto, Enrico Vicario, and Daniele Zingoni. "Sequence Retrieval by Contents through Spatio Temporal Indexing." In: Proceedings of 1993 IEEE Symposium on Visual Languages in Bergen, Norway, IEEE Computer Society Press, 88-92, 1993.
- Del Bimbo, Alberto, Enrico Vicario, and Daniele Zingoni. "A Spatio-Temporal Logic for Image Sequence Coding and Retrieval." In: Proceedings of 1992 IEEE Workshop on Visual Languages in Seattle, Washington, IEEE Computer Society Press, 228-230, 1992.
- Delanay, Robert. "The Eiffel Tower." 1911.
- Deren, Maya. "At Land." USA: 1947.
- Derrida, Jacques. Of Grammatology. Translated by Gayatri Chakravorty Spivak. Baltimore, Maryland: The Johns Hopkins University Press, 1974.

- Don, Abbe. "Narrative and the Interface." In The Art of Human Computer Interface Design, ed. Brenda Laurel. Reading, Massachusetts: Addison-Wesley, 1990.
- Dreyfuss, Henry. Symbol Sourcebook: An Authoritative Guide to International Graphic Symbols. New York: McGraw-Hill, 1972.
- Dubrovsky, Ben. Personal Communication. 1991.
- Duchamp, Marcel. "Nude Descending a Staircase #2." 1912.
- Ducrot, Oswald and Tzvetan Todorov. Encyclopedic Dictionary of the Sciences of Language. Translated by Catherine Porter. Baltimore, Maryland: The Johns Hopkins University Press, 1979.
- Eco, Umberto. "Articulations of the Cinematic Code." In Movies and Methods: An Anthology, ed. Bill Nichols. 590-607. Berkeley: University of California Press, 1976a.
- Eco, Umberto. A Theory of Semiotics. Advances in Semiotics, ed. Thomas A. Sebeok. Bloomington: Indiana University Press, 1976b.
- Eisenstein, Sergei M. Eisenstein 2: A Premature Celebration of Eisenstein's Centenary. Translated by Alan Y. Upchurch, N. Lary, Zina Voynow, and Samuel Brody. ed. Jay Leyda. Calcutta: Seagull Books, 1985.
- Eisenstein, Sergei M. Film Essays and a Lecture. Princeton, New Jersey: Princeton University Press, 1982.
- Eisenstein, Sergei M. Notes of a Film Director. Translated by X. Danko. New York: Dover Publications, Inc., 1970.
- Eisenstein, Sergei M. Film Form: Essays in Film Theory. Translated by Jay Leyda. San Diego: Harcourt Brace Jovanovich, Publishers, 1949.
- Eisenstein, Sergei M. The Film Sense. Translated by Jay Leyda. San Diego: Harcourt Brace Jovanovich, Publishers, 1947.
- Eisenstein, Sergei M. "October." Moscow: Sovkino, 1928.
- Elliott, Edward Lee. "WATCH • GRAB • ARRANGE • SEE: Thinking with Motion Images via Streams and Collages." M.S.V.S. Thesis, Massachusetts Institute of Technology, 1993.
- Elsaesser, Thomas, ed. Early Cinema: Space, Frame, and Narrative. Bloomington, Indiana: Indiana University Press, 1990.
- Falbel, Aaron. Constructionism: Tools to build (and think) with. Billund, Denmark: The LEGO Group, 1993.

- Feiner, Steven K. and Kathleen R. McKeown. "Generating Coordinated Multimedia Explanations." In: Proceedings of Sixth IEEE Conference on Artificial Intelligence Applications in Santa Barbara, California, 1990.
- Fraase, Michael. Macintosh Hypermedia: Volume I, Reference Guide. Glenview, Illinois: Scott, Foresman and Company, 1990.
- Fuji, Hideo and Robert R. Korfhage. "Features and a Model for Icon Morphological Transformation." In: Proceedings of 1991 IEEE Workshop on Visual Languages in Kobe, Japan, IEEE Computer Society Press, 240-245, 1991.
- Furnas, G.W., T.K. Landauer, L.M. Gomez, and S.T. Dumais. "The Vocabulary Problem in Human-System Communication." Communications of the ACM 30 (11 1987): 964-971.
- Gardner, Helen, Horst de la Croix, Richard G. Tansey, and Diane Kirkpatrick. Gardner's Art through the Ages. 9th ed., Vol. 1. San Diego, California: Harcourt Brace Jovanovich, 1991.
- Gorbman, Claudia. Unheard Melodies: Narrative Film Music. Bloomington, Indiana: Indiana University Press, 1987.
- Goto, Masataka and Yoichi Muraoka. "A Beat Tracking System for Acoustic Signals of Music." In: Proceedings of ACM Multimedia 1994 in San Francisco, California, ACM Press, 364-372, 1994.
- Greenway, Tom and Ronaldo Mouchawar. "A Visual Language for the Management of Digital Film and Video (Motion Imagery) Archives." In: Proceedings of 136th Technical Conference and World Media Expo of the Society of Motion Picture and Television Engineers (SMPTE) in Los Angeles, California, edited Society of Motion Picture and Television Engineers, Inc., 1-15, 1994.
- Groenewegen-Frankfort, Henriette Antonia. Arrest and Movement : An Essay on Space and Time in the Representational Art of the Ancient Near East. Cambridge, Massachusetts: Belknap Press, 1987.
- Gruber, H. and Voneche and J.J., ed. The Essential Piaget. New York: Basic Books, Inc., 1986.
- Guest, Ann Hutchinson. Dance Notation: the process of recording movement on paper. Brooklyn, New York: Dance Horizons, 1984.
- Guha, Ramanathan V. Personal Communication. 1994.
- Guha, Ramanathan V. and Douglas B. Lenat. "Enabling Agents to Work Together." Communications of the ACM 37 (7 1994): 127-142.

- Haase, Ken. "FRAMER: A Persistent Portable Representation Library." In: Proceedings of European Conference on Artificial Intelligence in Amsterdam, The Netherlands, 1994.
- Haase, Ken. "Integrating Analogical and Case-Based Reasoning in a Dynamic Memory." Internal Document. Cambridge, Massachusetts: MIT Media Laboratory, 1993.
- Haase, Ken. "Making Clouds from Cement: Building Abstractions out of Concrete Examples." In: Proceedings of US-Japan Workshop on Integrated Comprehension and Generation in Perceptually Grounded Environments in Japan, 1991.
- Haase, Ken and Warren Sack. "FRAMER Manual." Internal Document. Cambridge, Massachusetts: MIT Media Laboratory, 1993.
- Hampapur, Arun, Ramesh Jain, and Terry Weymouth. "Digital Video Segmentation." In: Proceedings of ACM Multimedia 1994 in San Francisco, California, ACM Press, 357-364, 1994.
- Haraway, Donna J. "A Cyborg Manifesto: Science, Technology, and Socialist-Feminism in the Late Twentieth Century." In Simians, Cyborgs, and Women: The Reinvention of Nature, 151-181. London: Free Association Books, 1991.
- Hawkes, Terence. Structuralism and Semiotics. New Accents, ed. Terence Hawkes. Berkeley: University of California Press, 1977.
- Hawley, Michael. "Structure out of Sound." Ph.D. Thesis, Massachusetts Institute of Technology, 1993.
- Hayes, Patrick J. "The Logic of Frames." In Readings in Knowledge Representation, ed. Ronald J. Brachman and Hector J. Levesque. 287-295. San Mateo, California: Morgan Kaufmann Publishers, 1985.
- Hecht, Werner, ed. Bertolt Brecht Schriften: Über Theater. Berlin: Henschelverlag Kunst und Gesellschaft, 1977.
- Heidegger, Martin. "Der Ursprung des Kunstwerkes." In Holzwege, 1-72. 6th ed., Frankfurt am Main: Vittorio Klostermann, 1980.
- Heidegger, Martin. Sein and Zeit. 15th ed., Tübingen, Germany: Max Niemeyer Verlag, 1979.
- Hitchcock, Alfred. "North by Northwest." USA: 1959.
- Hochberg, Julian E. "Representation of Motion and Space in Video and Cinematic Displays." In Handbook of Perception and Human Performance: Sensory Processes and Perception, ed. Kenneth R. Boff, Lloyd Kaufman, and James P. Thomas. Vol. 1. New York: John Wiley, 1986.

- Horn, Berthold K. P. and Brian G. Schmuck. "Determining Optical Flow." Artificial Intelligence 17 (1981): 185-203.
- Horton, William. The Icon Book: Visual Symbols for Computer Systems and Documentation. New York: John Wiley & Sons, Inc., 1994.
- Isehour, John Preston. "The Effects of Context and Order in Film Editing." AV Communications Review 23 (1 1975): 69-80.
- Iser, Wolfgang. The Fictive and the Imaginary: Charting Literary Anthropology. Baltimore, Maryland: The Johns Hopkins University Press, 1993.
- Iser, Wolfgang. "The Play of the Text." In Prospecting: From Reader Response to Literary Anthropology, 249-309. Baltimore, Maryland: The Johns Hopkins University Press, 1989a.
- Iser, Wolfgang. Prospecting: From Reader Response to Literary Anthropology. Baltimore, Maryland: The Johns Hopkins University Press, 1989b.
- Iser, Wolfgang. The Act of Reading: A Theory of Aesthetic Response. Baltimore, Maryland: The Johns Hopkins University Press, 1978.
- Iser, Wolfgang. The Implied Reader: Patterns of Communication in Prose Fiction from Bunyan to Beckett. Baltimore, Maryland: The Johns Hopkins University Press, 1974.
- Jenkins, Henry. Textual Poachers: Television Fans & Participatory Culture. Studies in Culture and Communication, ed. John Fiske. New York: Routledge, 1992.
- Jones, Chuck. "Music and the Animated Cartoon." Hollywood Quarterly 1:4 (July 1946): 364-370.
- Kahn, Kenneth. Creation of Computer Animations from Story Descriptions. Massachusetts Institute of Technology Artificial Intelligence Laboratory, 1979. Technical Report 540.
- Karp, Peter and Steven Feiner. "Issues in the Automated Generation of Animated Presentations." In: Proceedings of Graphics Interface '90 in Halifax, 39-48, 1990.
- Katz, Steven D. Film Directing Shot By Shot: Visualizing from Concept to Screen. Studio City, California: Michael Wiese Productions, 1991.
- Kern, Stephen. The Culture of Time and Space: 1880-1918. Cambridge, Massachusetts: Harvard University Press, 1983.

- Korfhage, Robert R. and Margaret A. Korfhage. "Criteria for Iconic Languages." In Visual Languages, ed. Shi-Kuo Chang, Tadao Ichikawa, and Panos A. Ligomenides. 207-231. New York: Plenum Press, 1986.
- Kornel, Amiel Mark. "Emerging Markets for Visual Media." M.S. Thesis, Massachusetts Institute of Technology Sloan School of Management, 1992.
- Kuleshov, Lev. "The Origins of Montage." In Cinema in Revolution, ed. Luda Schnitzer, Jean Schnitzer, and Marcel Martin. 67-76. New York: Da Capo Press, 1973.
- Kuleshov, Lev. Kuleshov on Film: Writings by Lev Kuleshov. Translated by Ronald Levaco. Berkeley: University of California Press, 1974.
- Lakoff, George. Women, Fire, and Dangerous Things: What Categories Reveal about the Mind. Chicago: The University of Chicago Press, 1987.
- Lasky, Alan. "Slipstream: A Data Rich Production Environment." M.S. Thesis, Massachusetts Institute of Technology, 1990.
- Laurel, Brenda, Tim Oren, and Abbe Don. "Issues in Multimedia Interface Design: Media Integration and Interface Agents." In: Proceedings of CHI '90, 133-139, 1990.
- Léger, Fernand and Dudley Murphy. "Ballet mécanique." 1924.
- Lenat, Douglas B. Personal Communication. 1994a.
- Lenat, Douglas B. "Strongly Semantic Information Retrieval." Invited Talk at AAAI-94 Workshop on Indexing and Reuse in Multimedia Systems in Seattle, Washington, 1994b.
- Lenat, Douglas B. Personal Communication. 1993.
- Lenat, Douglas B. and Ramanathan V. Guha. "Strongly Semantic Information Retrieval." In: Proceedings of AAAI-94 Workshop on Indexing and Reuse in Multimedia Systems in Seattle, Washington, edited by Catherine Baudin, Marc Davis, Smadar Kedar, and Daniel M. Russell, AAAI Press, 58-68, 1994.
- Lenat, Douglas B. and Ramanathan V. Guha. Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc., 1990.
- Levin, Golan. "Media Streams 3.0: User's Guide and Manual." ed. Marc Davis. Internal Document. Palo Alto, California: Interval Research Corporation, 1994.

- Levitt, David and Glorianna Davenport. "Symbolic Description of Movie Media." Unpublished NSF Proposal. Cambridge, Massachusetts: MIT Media Laboratory, 1987.
- Leyda, Jay. KINO: A History of the Russian and Soviet Film: A Study of the Development of Russian Cinema from 1896 to the Present. Princeton University Press, 1973.
- London, Kurt. Film Music. Translated by Eric S. Bensinger. London: Faber & Faber, 1936.
- M. V. D. "Tapestry." USA: 1990.
- Mackay, Wendy E. and Glorianna Davenport. "Virtual Video Editing in Interactive Multimedia Applications." Communications of the ACM 32 (7 1989): 803-110.
- MacNeil, Ron. "Capturing Multimedia Design Knowledge Using TYRO, the Constraint Based Designer's Apprentice." In: Proceedings of SPI/SPSE Symposium on Electronic Imaging in Cambridge, MIT Press, 1-9, 1991a.
- MacNeil, Ron. "Generating Multimedia Presentations Automatically Using TYRO: the Constraint, Case-Based Designer's Apprentice." In: Proceedings of 1991 IEEE Workshop on Visual Languages in Kobe, Japan, IEEE Computer Society Press, 74-79, 1991b.
- MacNeil, Ron. "Adaptive Perspectives: Case-Based Reasoning with TYRO, the Graphic Designer's Apprentice." In: Proceedings of IEEE 1990 Workshop on Visual Languages. MIT Press, 1990.
- MacNeil, Ron. "TYRO: A Constraint Based Graphic Designer's Apprentice." In: Proceedings of IEEE 1989 Workshop on Visual Language in edited MIT Press, 1989.
- McCarthy, John and Patrick Hayes. "Some Philosophical Problems from the Standpoint of Artificial Intelligence." In Machine Intelligence 4, Edinburgh: Edinburgh University Press, 1969.
- McCarthy, John. "Programs with Common Sense." In: Proceedings of Symposium on the Mechanization of Thought Processes in National Physical Laboratory, Teddington, England, 1958.
- McCloud, Scott. Understanding Comics: The Invisible Art. Northampton, Massachusetts: Tundra Publishing, 1993.
- McLuhan, Marshall. The Gutenberg Galaxy: The Making of Typographic Man. Toronto: University of Toronto Press, 1962.
- Meehan, James. "The Metanovel: Writing Stories by Computer." Ph.D. Thesis, Yale University, 1976.

- Merleau-Ponty, Maurice. Phenomenology of Perception. Translated by Colin Smith. International Library of Philosophy and Scientific Method, ed. Ted Honderich. London: Routledge & Kegan Paul, 1962.
- Metz, Christian. "Aural Objects." Yale French Studies 60 (1980): 24-32.
- Metz, Christian. "Current Problems of Film Theory: Mitry's *L'Esthetique et Psychologie du Cinema, Vol. II.*" In Movies and Methods: An Anthology, ed. Bill Nichols. 568-578. Berkeley: The University of California Press, 1976.
- Metz, Christian. Film Language: A Semiotics of Cinema. Translated by Michael Taylor. Chicago: The University of Chicago Press, 1974.
- Michelson, Annette, ed. Kino-Eye: The Writings of Dziga Vertov. Berkeley: University of California Press, 1984.
- Mills, Michael, Jonathan Cohen, and Yin Yin Wong. "A Magnifier Tool for Video Data." In: Proceedings of CHI'92 in Monterey, California, 93-98, 1992.
- Minsky, Marvin. A Framework for Representing Knowledge. Massachusetts Institute of Technology, 1974. Memo No. 206.
- Minsky, Marvin. Society of Mind. New York: Simon and Shuster, 1987.
- Morgenroth, Lee H. "Homer: A Video Story Generator." B.S. Thesis, Massachusetts Institute of Technology, 1992.
- Musser, Charles. The Emergence of Cinema: The American Screen to 1907. Vol. 1. History of the American Cinema, ed. Charles Harpole. Berkeley: University of California Press, 1990.
- Nagasaka, Akio and Yuzuru Tanaka. "Automatic Video Indexing and Full-Video Search for Object Appearances." In IFIP Transactions, Visual Database Systems II, ed. E. Knuth and L. M. Wegner. Elsevier Publishers, 1992.
- Neurath, Otto. International Picture Language. New York: State Mutual Book and Periodical Service, 1981.
- Newell, Alan, J. C. Shaw, and Herbert A. Simon. "GPS: A Program That Simulates Human Thought." In Computers and Thought, ed. Edward A. Feigenbaum and Julian Feldman. 279-293. New York: McGraw-Hill, 1963.
- Oren, Tim, Gitta Salomon, Kristee Kreitman, and Abbe Don. "Guides: Characterizing the Interface." In The Art of Human Computer Interface Design, ed. Brenda Laurel. Reading, Massachusetts: Addison-Wesley, 1990.

- Osgood, Richard. "Question-Based Conceptual Indexing of Conversational Multimedia." In: Proceedings of AAAI-94 Workshop on Indexing and Reuse in Multimedia Systems in Seattle, Washington, edited by Catherine Baudin, Marc Davis, Smadar Kedar, and Daniel M. Russell, AAAI Press, 141-150, 1994.
- Otsuji, Kiyotaka, Yoshinobu Tonomura, and Yuji Ohba. "Video Browsing Using Brightness Data." SPIE Visual Communications and Image Processing '91: Image Processing SPIE 1606 (1991): 980-989.
- Papert, Seymour. Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books, Inc., 1980.
- Parkes, Alan P. "An Artificial Intelligence Approach to the Conceptual Description of Images." Ph.D. Thesis, Lancaster University, 1989a.
- Parkes, Alan P. "Settings and the Settings Structure: The Description and Automated Propagation of Networks for Perusing Videodisk Image States." In: Proceedings of Twelfth International Conference on Research and Development in Information Retrieval (SIGIR '89) in Cambridge, Massachusetts, edited by N. J. Belkin and C. J. van Rijsbergen, 229-238, 1989b.
- Pentland, Alex P., Rosalind Picard, Glorianna Davenport, and Kenneth Haase. Video and Image Semantics: Advanced Tools for Telecommunications. Massachusetts Institute of Technology Media Laboratory, 1994a. Perceptual Computing Technical Report 283.
- Pentland, Alex P., Rosalind W. Picard, and Stan Sclaroff. Photobook: Tools for Content-Based Manipulation of Image Databases. Massachusetts Institute of Technology Media Laboratory, 1994b. Perceptual Computing Technical Report 255.
- Pentland, Alex P., N. Etcoff, A. Masoiu, O. Oliyide, Thad Starner, and Matthew Turk. "Experiments with Eigenfaces." In: Proceedings of Looking At People Workshop at IJCAI '93 in Chamberry, France, 1993a.
- Pentland, Alex P., Rosalind Picard, Glorianna Davenport, and Kenneth Haase. The BT/MIT Project on Advanced Image Tools for Telecommunications: An Overview. Massachusetts Institute of Technology Media Laboratory, 1993b. Perceptual Computing Technical Report 212.
- Picard, Rosalind and Fang Liu. "A New World Order for Image Similarity." In: Proceedings of IEEE Conference on Acoustics, Speech, and Signal Processing in Adelaide, Australia, 1994.
- Pincever, Natalio C. "If You Could See What I Hear: Editing Assistance Through Cinematic Parsing." M.S. Thesis, Massachusetts Institute of Technology, 1990.

- Prelinger, Richard. "ARCHIVAL SURVIVAL: The Fundamentals of Using Film Archives and Stock Footage Libraries." The Independent Film & Video Monthly, 1991.
- Prendergast, Roy M. Film Music: A Neglected Art. New York: W. W. Norton and Company, 1977.
- Pudovkin, Vsevolod Illarionovitch. Film Technique and Film Acting. Translated by Ivor Montagu. New York: Bonanza Books, 1949.
- Rappaport, Mark. "Rock Hudson's Home Movies." USA: 1993.
- Read, Gardner. Music Notation: A Manual of Modern Practice. 2nd ed., Boston: Allyn and Bacon, 1969.
- Reynolds, Douglas A. "A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification." Ph.D. Thesis, Georgia Institute of Technology, 1992.
- Reynolds, Douglas A. "Speaker Identification and Verification using Gaussian-mixture Speaker Models." In: Proceedings of ESCA Workshop on Automatic Speaker Recognition, Identification and Verification '94, Georgia Institute of Technology, 27-30, 1994.
- Riesbeck, Charles and Roger C. Schank. Inside Case-Based Reasoning. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1989.
- Rosenblum, Ralph and Robert Karen. When The Shooting Stops, The Cutting Begins - A Film Editor's Story. New York: Da Capo Paperback, 1979.
- Rosenthal, David. "Machine Rhythm: Computer Emulation of Human Rhythm Perception." Ph.D. Thesis, Massachusetts Institute of Technology, 1992.
- Rouse, Russell. "The Thief." USA: 1952.
- Rubin, Benjamin and Glorianna Davenport. "Structured Content Modeling for Cinematic Information." In: Proceedings of Workshop on Video as a Research and Design Tool in Cambridge, Massachusetts, MIT Media Laboratory, 1989.
- Rubin, Benjamin. "Constraint-Based Cinematic Editing." M.S.V.S. Thesis, Massachusetts Institute of Technology Media Laboratory, 1989.
- Rubin, Michael. NonLinear: A Guide to Electronic Film and Video Editing. Gainesville, Florida: Triad Publishing Company, 1991.

- Sack, Warren and Marc Davis. "IDIC: Assembling Video Sequences from Story Plans and Content Annotations." In: Proceedings of IEEE International Conference on Multimedia Computing and Systems in Boston, Massachusetts, IEEE Computer Society Press, 30-36, 1994.
- Sack, Warren. Actor-Role Analysis: Ideology, Point of View, and the News. Massachusetts Institute of Technology Media Laboratory Learning and Common Sense Section, 1994a. Tech Report 94-005.
- Sack, Warren. "On the Computation of Point of View." In: Proceedings of Twelfth National Conference on Artificial Intelligence (AAAI-94) in Seattle, Washington, 1488, 1994b.
- Sampson, Geoffrey. Writing Systems: A Linguistic Introduction. Stanford, California: Stanford University Press, 1985.
- Sasnett, Russell. "Reconfigurable Video." M.S.V.S. Thesis, Massachusetts Institute of Technology, 1986.
- Saussure, Ferdinand de. Course in General Linguistics. Translated by Wade Baskin. New York: McGraw-Hill, 1983.
- Schank, Roger C. Personal Communication. 1993.
- Schank, Roger C. and Charles J. Rieger. "Inference and the Computer Understanding of Natural Language." In Readings in Knowledge Representation, ed. Ronald J. Brachman and Hector J. Levesque. 120 - 139. San Mateo, California: Morgan Kaufmann Publishers, Inc., 1985.
- Schank, Roger C. and Charles Riesbeck. Inside Computer Understanding: Five Programs Plus Miniatures. The Artificial Intelligence Series, ed. Roger C. Schank. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1981.
- Schank, Roger C. Dynamic Memory: A Theory of Reminding and Learning in Computers and People. Cambridge: Cambridge University Press, 1982.
- Seifert, Colleen, Lee Brooks, Mark Burstein, Dedre Gentner, Brian Ross, and Manuela Veloso. "Panel on "Analogy and CBR"." In: Proceedings of Case-Based Reasoning Workshop in Pensacola Beach, Florida, edited by Kristian Hammond, Morgan Kaufmann Publishers, Inc., 125-158, 1989.
- Shub, Esfir. "The Fall of the Romanov Dynasty." Moskow: 1927.
- Siporin, Sheldon. "When Fair is Foul: Fair Use and Copyright." The Independent, 1990, 20-24.

- Smoliar, Stephen W., Hongjang Zhang, and J. H. Wu. "Using Frame Technology to Manage Video." In: Proceedings of AAAI-94 Workshop on Indexing and Reuse in Multimedia Systems in Seattle, Washington, edited by Catherine Baudin, Marc Davis, Smadar Kedar, and Daniel M. Russell, 1994.
- Stam, Robert, Robert Burgoyne, and Sandy Flitterman-Lewis. New Vocabularies in Film Semiotics: Structuralism, Post-Structuralism, and Beyond. Sightlines, ed. Edward Buscombe and Phil Rosen. London: Routledge, 1992.
- Star Trek: The Next Generation. "The Ensigns of Command." Los Angeles, California: Paramount Television, 1989.
- Strickland, Rachel and Jill Wright. Backyard Transformations: An Interim Project Report. Internal Document. Apple Vivarium Program, 1990.
- Suleiman, Susan R. and Inge Crosman, ed. The Reader in the Text. Princeton: Princeton University Press, 1980.
- Swanberg, Deborah, Chiao-Fe Shu, and Ramesh Jain. "Knowledge-Guided Parsing in Video Databases." In: Proceedings of SPIE 1908: Storage and Retrieval for Image and Video Databases in San Jose, edited by Wayne Niblack, 13-24, 1993.
- Tanimoto, Steven L. and Marcia S. Runyan. "PLAY: An Iconic Programming System for Children." In Visual Languages, ed. Shi Kuo Chang, Tadao Ichikawa, and Panos A. Ligomenides. 191-205. New York: Plenum Press, 1986.
- Taylor, Richard and Ian Christie, ed. The Film Factory: Russian and Soviet Cinema in Documents. Harvard Film Studies. Cambridge, Massachusetts: Harvard University Press, 1988.
- Teodosio, Laura. "Salient Stills." M.S.V.S. Thesis, Massachusetts Institute of Technology, 1992.
- Thomson, Patricia. "Sleuth: The Search for Television News Footage." The Independent, 1988, 20-27.
- Tomasi, Carlo and Takeo Kanade. "Shape and Motion from Image Streams under Orthography: A Factorization Method." IJCV 9 (2 1992).
- Tompkins, Jane. Reader-Response Criticism. Baltimore, Maryland: The Johns Hopkins University Press, 1980.

- Tonomura, Yoshinobu, Akihito Akutsu, Kiyotaka Otsuji, and Toru Sadakata. "VideoMAP and VideoSpacelcon: Tools for Anatomizing Content." In: Proceedings of INTERCHI'93 Conference on Human Factors in Computing Systems in Amsterdam, The Netherlands, edited by Stacey Ashlund, Kevin Mullet, Austin Henderson, Erik Hollnagel, and Ted White, ACM Press, 131-136, 1993.
- Travers, Mike and Marc Davis. "Programming with Characters." In: Proceedings of 1993 International Workshop on Intelligent User Interfaces in Orlando, Florida, edited by Wayne D. Gray, William E. Hefley, and Dianne Murray, ACM Press, 269-272, 1993.
- Tufte, Edward R. Envisioning Information. Cheshire, Connecticut: Graphics Press, 1990.
- Tulving, Endel. "What is Episodic Memory?" Current Directions in Psychological Science 2 (3 1993): 67-70.
- Ueda, Hirotada, Takafumi Miyatake, and Satoshi Yoshizawa. "IMPACT: An Interactive Natural-Motion-Picture Dedicated Multimedia Authoring System." In: Proceedings of CHI '91 in New Orleans, Louisiana, ACM Press, 343-350, 1991.
- Ueda, Hirotada, Takafumi Miyatake, Shigeo Sumino, and Akio Nagasaka. "Automatic Structure Visualization for Video Editing." In: Proceedings of INTERCHI'93 Conference on Human Factors in Computing Systems in Amsterdam, The Netherlands, edited by Stacey Ashlund, Kevin Mullet, Austin Henderson, Erik Hollnagel, and Ted White, ACM Press, 137-141, 1993.
- Vertov, Dziga. "Man with a Movie Camera." Moskow: 1928.
- Vinge, Vernor. Personal Communication. 1994.
- Walker, Alexander. The Shattered Silents: How the Talkies Came to Stay. New York: Morrow, 1979.
- Weber, Karon. "The Work Practice of Moving Image Indexing." Invited Talk at AAAI-94 Workshop on Indexing and Reuse in Multimedia Systems in Seattle, Washington, 1994.
- Weiss, Ron, Andrzej Duda, and David K. Gifford. "Content-Based Access to Algebraic Video." In: Proceedings of IEEE International Conference on Multimedia Computing and Systems in Boston, Massachusetts, IEEE Computer Society Press, 140-151, 1994.
- Williams, Brian. "The Movie Map." B.S. Thesis, Massachusetts Institute of Technology, 1994.

- Winograd, Terry and Fernando Flores. Understanding Computers and Cognition: A New Foundation for Design. Norword, New Jersey: Ablex Publication Corp., 1986.
- Winograd, Terry. "Frame Representations and the Declarative/Procedural Controversy." In Readings in Knowledge Representation, ed. Ronald J. Brachman and Hector J. Levesque. 357-370. San Mateo, California: Morgan Kaufmann Publishers, 1985.
- Winograd, Terry. "On Primitives, Prototypes and Other Anomalies." In: Proceedings of Second Conference on Theoretical Issues in Natural Language Processing in Champaign-Urbana, Illinois, 1978.
- Wittgenstein, Ludwig. Philosophische Untersuchungen. Frankfurt am Main: Suhrkamp Taschenbuch Verlag, 1977.
- Woodfill, John. "Motion Vision and Tracking for Robots in Dynamic, Unstructured Environments." Ph.D. Thesis, Stanford University, 1992.
- Zabih, Ramin, John Woodfill, and Meg Withgott. "A Real-Time System for Automatically Annotating Unstructured Image Sequences." In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics in Le Touquet, France, IEEE Press, 345-350, 1993.
- Zhang, HongJiang and Stephen William Smoliar. "Developing Power Tools for Video Indexing and Retrieval." In: Proceedings of SPIE '94, 1994.
- Zhang, HongJiang, Yihong Gong, Stephen William Smoliar, and Shuang Yeo Tan. "Automatic Parsing of News Video." In: Proceedings of IEEE International Conference on Multimedia Computing and Systems in Boston, Massachusetts, IEEE Computer Society Press, 45-54, 1994.
- Zhang, HongJiang, Atreyi Kankanhalli, and Stephen William Smoliar. "Automatic Partitioning of Full-Motion Video." Multimedia Systems 1 (1993): 10-28.

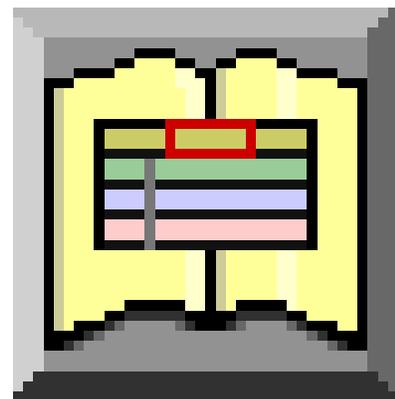
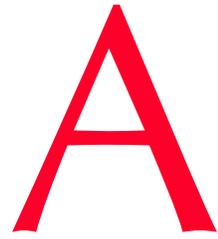


Appendix A

Media Streams User's Guide and Manual

Media Streams 3.0

**A Platform for Content-Based
Annotation and Retrieval
of Digital Video and Audio**



User's Guide and Manual

Using this Manual

This Manual for the Media Streams system has three parts — one which describes the functionality of the system's interface features, one which explains how to use them, and one which details the annotation theory and philosophy recommended by the system's developers.

The first part, **Overview of the System Components**, explains what the interface components are, what they look like, where to find them, what they're used for, what they allow you to do, and how they're organized. Use this section if you're not sure what a certain part of the interface is for. This section can help you get acquainted with Media Streams, and alert you to some of the system's functions that may not be immediately obvious.

The second part, **Using the System**, is intended as a reference guide for occasions when you know what operation you want to do, but you're unclear about how to do it.

Following the second part are several important appendices, including a glossary, a flow chart of a recommended work flow, and a table detailing the consequences of dragging icons from one place to another.

The third part, **Suggestions for Annotators**, is especially important — it details various issues you should keep in mind when annotating a Movie, and it outlines some of the precautions you should take to ensure the repurposability of your annotations.

- **If you want to get started logging right away, you will need to read the sections on:**

- The Icon Space and Media Time Line Diagrams (p.365, p.355)
- Annotator's Work Flow (p.403)
- Getting Around the Movie (p.381)
- Operations with the Workshop (p.385)
- Creating a Compound Icon (p.387)
- Creating a Glommed Icon (p.387)
- Operations with Annotations (p.389)

- **If you have more time, you should also read:**

- Descriptor Hierarchies (p.372)
- Suggestions for Annotators (p.405)
- Hiding a Stream (p.384)
- Making a Simple Query (p.392)
- Some Representative Queries (p.394)
- The Settings Palette (p.364)

Contents



Overview of the System Components

The Media Time Line

- The Movie Controls
- The Select Bar
- Streams
- The Movie Streams and their Thumbnails
- The Minutes and Seconds Scrubbers
- Stream Controls
- Time-Index Displays
- The Hidebars
- Annotations
- Compound icons
- Select Bar Icons
- The Settings Palette

The Icon Space

- Icons
 - Workshop Icons
 - Compound icons
 - Icon Titles
 - Media Time Line Icons
- The Icon Workshop
- Descriptor Hierarchies
- The Icon Information Editor
- The Animated Icon Editor
- The Icon Palettes



Using the System

Getting Around the Movie

- Getting Around the Movie with the Movie Controls
- Content Navigation with the Movie Controls
- Getting Around the Movie with the Thumbnails
- Getting Around the Movie with the Scrubbers
- Getting Around the Movie with the Select Bar

Operations with Streams

- Moving Streams Around
- Selecting and De-Selecting a Stream
- Hiding a Stream
- Recalling a Stream from the Hidebar
- Navigating through the Movie by Content
- Saving Your Annotated Media Time Line

Operations with the Workshop

- Navigating the Workshop
- Bubble Help
- Filtering the Workshop with an Icon from the Media Time Line or Icon Palette

Operations With Icons

- The Compounding Modes: What They Mean
- Creating a Compound Icon
- Creating a Glommed Icon
- Creating a Compound Transition Icon
- Viewing, Declaring or Changing an Icon's Information

Operations with Annotations

- Making an Annotation on the Media Time Line
- Cropping an Annotation
- Adjusting the End-Points of an Annotation
- Removing an Annotation
- Moving Annotations Around
- Undo

Operations with the Icon Palette

- Making a Simple Query
- The Query Language
- Some Representative Queries
- Sorting Your Results
- Sorts and Orders of Operations

A Media Streams Glossary

Annotator's Work Flow

Dragging Icons: an Appendix



Suggestions for Annotators

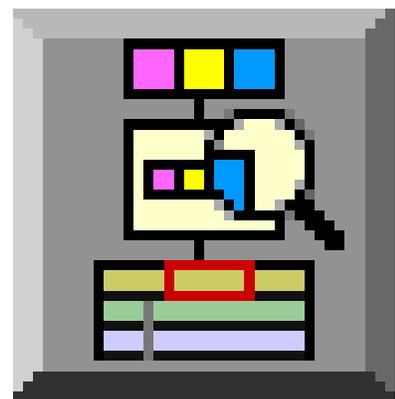
- The Laws of Logging**
- Inferable Time**
- Inferable Space**
- Weather**
- Timestamp**
- Spacestamp**
- Characters**
- Objects**
- Relative Positions**
- Character Actions**
- Object Actions**
- Cinematography**

Table of Figures

Fig. 96.	The Media Time Line
Fig. 97.	The Movie Controls
Fig. 98.	The Select Bar
Fig. 99.	A non-expandable stream and an expandable stream
Fig. 100.	The three Movie Streams and the amount of the Movie they can represent at any one time
Fig. 101.	A segment of a Seconds Thumbnails stream
Fig. 102.	A segment of a Videogram
Fig. 103.	The Minutes Scrubber and the Seconds Scrubber
Fig. 104.	The Cinematography Stream Controls and their hierarchy
Fig. 105.	The Time-Index displays
Fig. 106.	The Audio and Video Hidebars
Fig. 107.	The Elements of an Annotation
Fig. 108.	The Icon Space
Fig. 109.	Media Time Line Icons
Fig. 110.	The Icon Workshop
Fig. 111.	The Icon Information Editor and the Icon Title Editor
Fig. 112.	The Animated Icon Editor
Fig. 113.	An Icon Palette
Fig. 114.	Filter units can return Media Time Line Icons
Fig. 115.	Typical Work Flow in Annotation
Fig. 116.	Functional Space versus scenery
Fig. 117.	Actual versus Inferable Location
Fig. 118.	Objects at different focal distances can have identical Framings

*User's Guide and Manual written and designed by
Golan Levin, August 1994.*

Part One



Overview of the System Components

The Media Streams Media Time Line

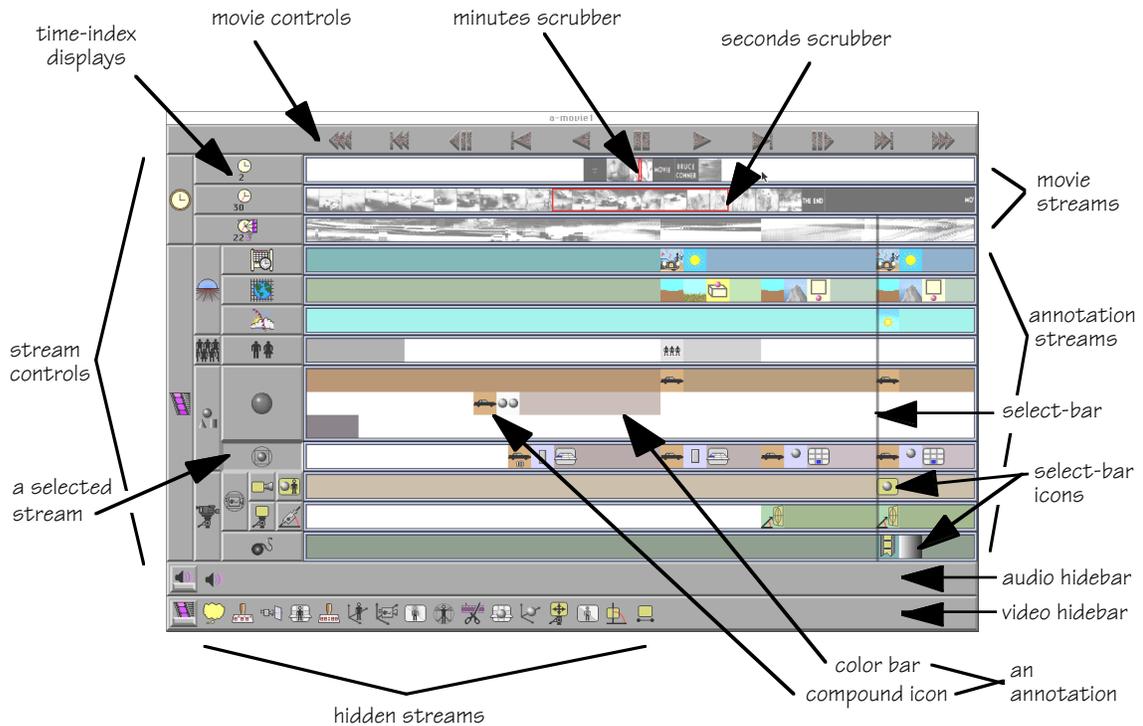


Fig. 96. The Media Time Line

The Media Time Line

The Media Time Line is an interface for creating, browsing, and editing temporally indexed annotations of digital media. Annotations appear as colored stripes that are arranged in horizontal tracks called streams. Each stream's annotations are dedicated to describing a specific aspect of the digital Movie, such as the Movie's characters or cinematography. The Media Time Line provides tools for re-arranging and hiding streams, like the Stream Controls and the Hidebars. It also provides tools, such as the Movie Controls, Movie Thumbnails, Movie Scrubbers, and the Select Bar, for navigating through a Movie. Finally, the Media Time Line provides a host of interface tools for the insertion, editing and removal of iconic annotations. Media Streams allows you to have more than one Media Time Line, and more than one Movie, open at the same time.

The Movie Controls

Media Streams provides eleven tape-deck style controls for navigating the Movie. Located in a horizontal bar at the top of the Media Time Line, these controls allow you to move through the Movie at a variety of speeds, in different directions, by shot boundaries, and even by changes in the Movie's content. Some of these controls, such as "Play," "Pause," and "Scan" are conventional controls for navigating tape media; others, such as "Frame Forward" are specific to video and digital video; and others, such as "Play Extent" and "Jump" have been created specifically for use with annotated media. For a detailed description of their functionality, see "Getting Around the Movie with the Movie Controls" (p.381).



Fig. 97. The eleven Movie Controls. From left to right, they are: Scan Reverse, Jump Reverse, Frame Reverse, Reverse Play Extent, Reverse Play, Pause, Play, Play Extent, Frame Forward, Jump Forward, and Scan Forward.

The Select Bar

The thin gray line running vertically down the Media Time Line is the Select Bar — a multi-purpose interface for navigating video and manipulating annotations. It is a scrub tool, for moving through the Movie. It is also an indicator, registering the Movie's **current frame** on the Videogram and, on its right side, displaying an iconographic slice through the current frame's content. Finally, it is the primary tool for inserting and cropping annotations on the Time Line.

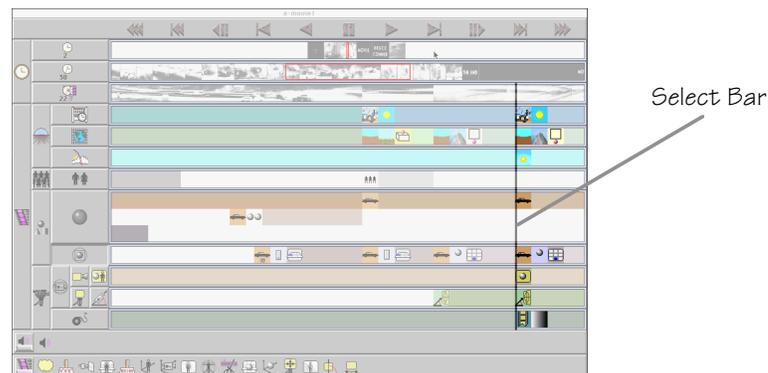


Fig. 98. The Select Bar is an indicator of both time and content: its horizontal position indicates the time of the current frame on the Videogram, while its vertical accessories describe the current frame. Reading down the Select Bar, we see that this shot depicts a clear day in the 1920's, set in front of a mountain; a car is falling and the camera, framing it in a medium shot, tracks it. The Select Bar is also a scrub tool and a tool for inserting and cropping annotations.

Streams

The broad horizontal stripes that occupy the greatest part of the Media Time Line are its **streams**. Nearly all of these streams, with a small number of exceptions, display iconic annotations describing the content of the Movie. Each stream's annotations are dedicated to a specific aspect of the Movie's content, such as the Movie's characters or cinematography. Up to 15 streams can be simultaneously displayed, though many more are hidden: the system defines streams for 44 different aspects of media content.

Media Streams organizes its annotation streams into **hierarchies**. The primary fork of the stream-hierarchy divides streams into those containing descriptions of the audio track, and those containing descriptions of the video track of the Movie. At first glance, it might appear as if there is a great deal of duplication between the audio annotation streams and the video annotation streams — both groups have streams for the description of characters, objects, and other aspects of media content. In fact, the audio and video annotation streams are not redundant at all: it is entirely possible for the video and audio tracks of a Movie to be wholly unrelated. **In Media Streams, the audio and video are logged separately.**

Further subdivisions of the stream-hierarchy organize streams into groups devoted to the annotation of characters, objects, cinematography, transitions, and other aspects of content. The group of streams concerned with characters, for example, contains streams for “characters,” “character actions,” “character screen positions,” and “character relative positions.” These groupings and divisions of the stream-hierarchy are mirrored in the graphical organization of the Stream Controls, which progress from **superordinate** stream-groups to **terminal** streams when read from left to right.

Annotation streams come in two varieties: **expandable** and **non-expandable**. Expandable streams are those which can have multiple simultaneous annotations. For example, a description of a dinner scene might contain simultaneous annotations for chairs, silverware, napkins and food — all in the “Objects” stream together. Non-expandable streams, on the other hand, can only have one annotation at any given time. It would be a logical contradiction, for example, for a shot's “Weather” to be overcast and clear simultaneously, or for a single shot's location be in New York and Antarctica at the same time. These streams, and others like “Time” and “Media Type,” do not permit more than one annotation at a time.

Streams or parts of streams that lack annotations are white, while the annotations that reside in the streams are represented by an icon followed by a stripe of color. Details about how to manipulate streams and annotations are given in the later sections, “Operations with Streams” and “Operations with Annotations.”

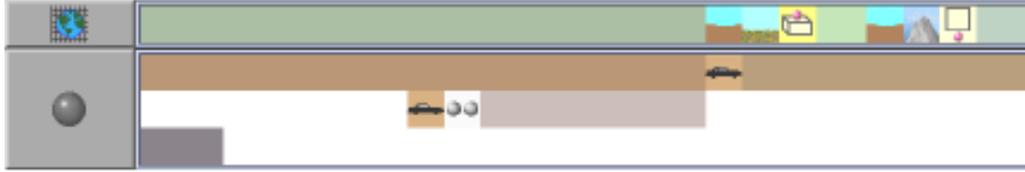


Fig. 99. A non-expandable stream (Inferable Time), at top, and an expandable one (Objects) below it. The expandable stream shown has expanded to a height of three annotations.

The Movie Streams and their Thumbnails

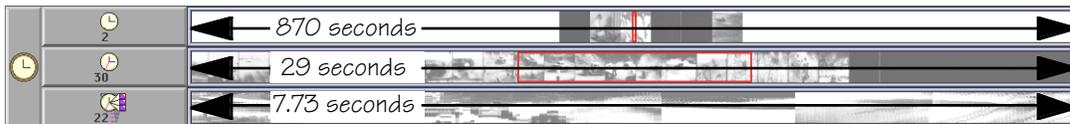


Fig. 100. The three Movie Streams and the amount of the Movie they can represent at any one time. The Minutes stream, at the top, displays 29 minutes of Movie-time; the Seconds stream (in the middle) displays 29 seconds of Movie-time, and the Videogram at bottom spans 232 frames of the Movie. These figures are for 21-inch, high-resolution monitors, and may vary with other configurations. A Movie Stream for “hours” is not presently supported due to the memory-expensive nature of digital video.

Three streams, known as the **Movie Streams**, offer special views onto the data of the Movie itself. Usually positioned at the top of the Media Time Line, these streams display actual samples of the Movie’s video. The upper two Movie Streams display Movie frame samples, called **Thumbnails**, which provide another convenient means of navigating the Movie: by double-clicking on them, you can jump through the Movie to the frame they represent. The first Movie Stream, known as the Minutes Thumbnails, shows thumbnails sampled from the Movie at regular intervals of one minute. The second Movie Stream displays thumbnails taken every second. The Minutes and Seconds Thumbnail streams provide an overview of the Movie, and, as such, are the only streams not displayed at the same time scale as the Annotation Streams.



Fig. 101. A segment of a Seconds Thumbnails stream, representing 13 seconds of video

Registering information at the level of frames (and to the same scale as the Annotation Streams) is the third Movie Stream, known as the **Videogram**. Instead of thumbnails, this stream displays narrow vertical strips

taken from the center of each Movie frame. The Videogram makes cuts and motion readily apparent, and thereby allows for quick inspection of the Movie. Specifically, shots taken with a moving camera, and/or containing moving subject matter, usually have wavy lines in their Videograms, while shots with more static images have Videograms with horizontal stripes. Cuts or “**shot boundaries**” generally appear as abrupt discontinuities in the Videogram.

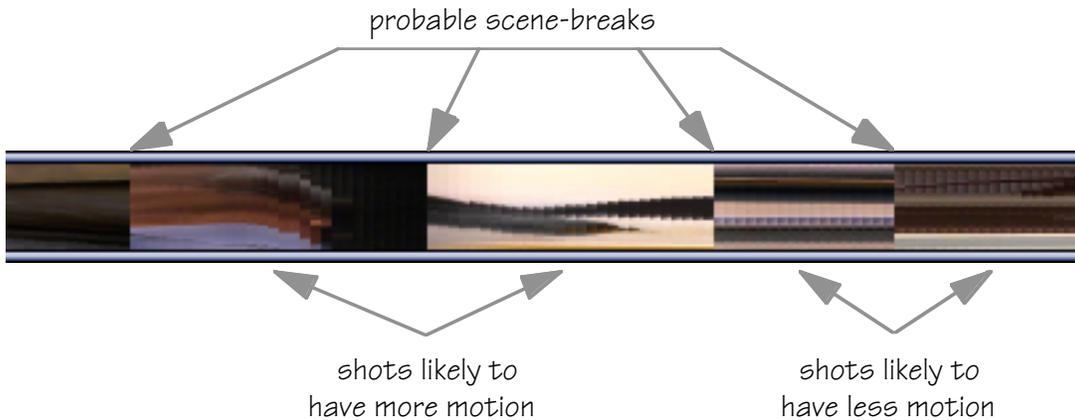


Fig. 102. A Videogram representing about 3.6 seconds of video

The Minutes and Seconds Scrubbers

Superimposed on the Minutes and Seconds Thumbnail streams are two red rectangles called the Minutes and Seconds **Scrubbers**. Clicking on and dragging these Scrubbers will allow you to move through the Movie at scales 225 and 3.75 times greater than the scale of the Select Bar and Annotation Streams. As you drag one of the Scrubbers, the Movie window displays what the new current frame will be when you release the Scrubber. Additionally, each scrubber windows the thumbnails beneath it, indexing with its edges where the visible range of the Annotation Streams begins and ends.

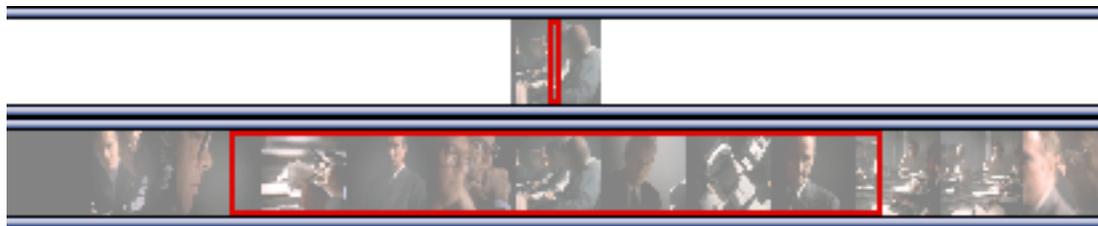


Fig. 103. The Minutes Scrubber, top, and the Seconds Scrubber

Stream Controls

Flanking the left edge of each stream are at least two beveled **Stream Controls**. These controls govern the arrangement and selection of the streams associated with them, and graphically display the streams' hierarchical organization. The Stream Controls furthest to the right are **terminal** Stream Controls, while the Stream Controls which group them are **superordinate**. Clicking and dragging on a Stream Control allows you to dynamically re-arrange the vertical position of its stream(s). Single-clicking on a Stream Control inverts it and **selects** the stream or group of streams associated with it; this selection determines, for instance, the stream(s) according to which you can navigate the video with the Jump and Play Extent Movie Controls. Double-clicking on a Stream Control hides the control and its stream or group of streams in one of the Hidebars at the bottom of the Media Time Line.

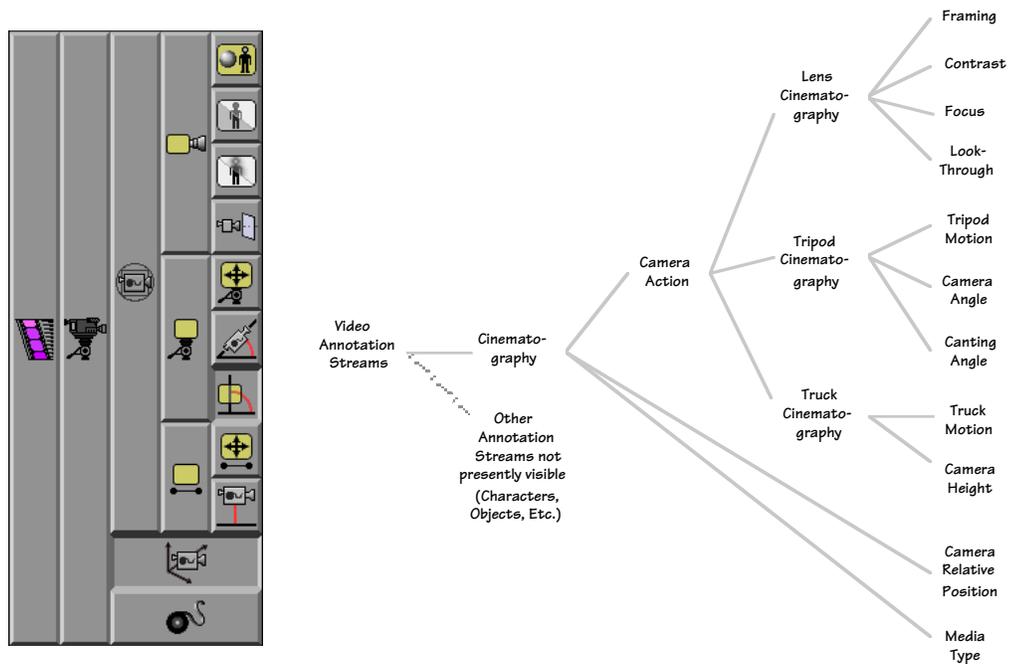
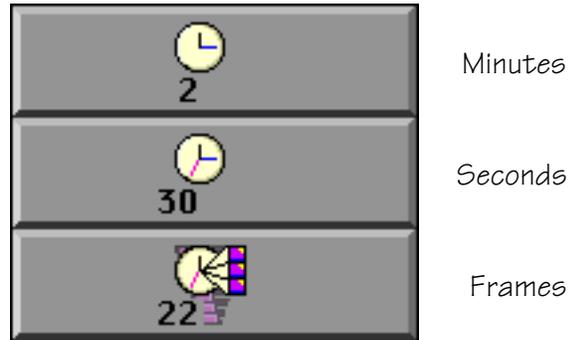


Fig. 104. On the left is the block of Cinematography Stream Controls; on the right is a diagram explicating the block's hierarchical organization. The complete hierarchy of Stream Controls for Cinematography has eleven terminal controls and five superordinate controls. Here, two of them (Canting Angle and Camera Relative Position) are shown in their selected state. Other Video Annotation stream-hierarchies on the same level as Cinematography, such as Characters and Objects, are not depicted.

Time-Index Displays

Located inside the Stream Controls of the Movie Streams are three small numbers which display the minute, second and frame number of the current frame. These numbers are displayed alongside, respectively, the Minutes Thumbnails, Seconds Thumbnails, and Videogram streams.

Fig. 105. The Time-Index displays for minutes, seconds and frames are located in the Stream Controls for their affiliated Movie Streams. Reading down, we see that the current frame is located 2 minutes, 30 seconds, and 22 frames into the Movie. A Time-Index display for “hours” is not presently supported due to the memory-expensive nature of digital video.



The Hidebars

Spanning the bottom of the Media Time Line are the Audio and Video Hidebars. Because the Media Streams Time Line can display no more than 15 of the 44 content streams it manages (on a 1024*768 pixel monitor), many streams are hidden on the Hidebars, where they are represented by their control-icons. Opening a hidden stream for display is accomplished by double-clicking on its Hidebar icon; hiding a Time Line stream is done by double-clicking on its Stream Control. The Hidebars and Stream Controls are your tools for dynamically managing which streams you want to see.



Fig. 106. The Audio and Video Hidebars. In this example, the Audio Hidebar contains the streams for Thoughts About the Audio, Audio Pause Bars, Audio Mise-en-Scene, Objects in the Audio, Characters in the Audio and Dialogue Transcription. Below it, the Video Hidebar contains the streams for Minutes Thumbnails, Inferable Space, Weather, Cinematography, Transitions, Characters Screen Position, Characters Relative Position, and Character Actions.

Annotations

Annotations are the graphical descriptors of the Movie's content, which appear in streams on the Media Time Line. Abstractly speaking, each annotation contains three pieces of information: **that** something occurs in the Movie, **when** it starts to occur, and **when** it ceases to be the case (i.e., for what duration of Movie-time is it true). This information is made visible in the annotation's two components: its **Compound icon** (a pictograph representing some aspect of the Movie's content, and the Media Time Line-location of its onset) and its **color bar** (a horizontal colored stripe which extends from the Compound icon to the Media Time Line-location of the content's termination). Annotations have a **start-frame**, indicated by the left edge of their Compound icons, and an **end-frame**, indicated by the right edge of their color-bars. The amount of the time for which the annotation is defined is indicated by the annotation's length, which is equal to the length of the Compound icon plus the length of the color bar. Put another way, the duration of the Movie for which the annotation is valid is bracketed by the annotation's start-frame and end-frames.

An annotation is constructed when an icon is dragged from the Icon Space Window to the Media Time Line. The icon is inserted at the select bar, becoming the Compound icon of the new annotation. Depending on how the Annotation Mode of the system is set, the color bar of the newly-created annotation will extend either to the next scene break (or to the next change in a selected stream), or indefinitely, to the end of the Movie. It then becomes your job to crop or tweak the end-frame of the annotation so that it does not extend past when it should. Methods for cropping annotations and changing their start-frames and end-frames are discussed later in the section, "Operations with Annotations"; the mechanics of creating and locating icons in the Icon Space are discussed in "Navigating the Workshop" and "Operations with the Icon Palette" (p.385, p.392).

The colors of annotations' color bars are determined by a function of the colors in their Compound icons. This feature can help you recognize annotations with just a glance at their color bars. For example, men and women have blue and pink character icons, respectively; it then becomes easy to visually discern the annotations of men and women because they are bluish and pinkish. The system will pick a different color for a new annotation if there is conflict within the same stream.

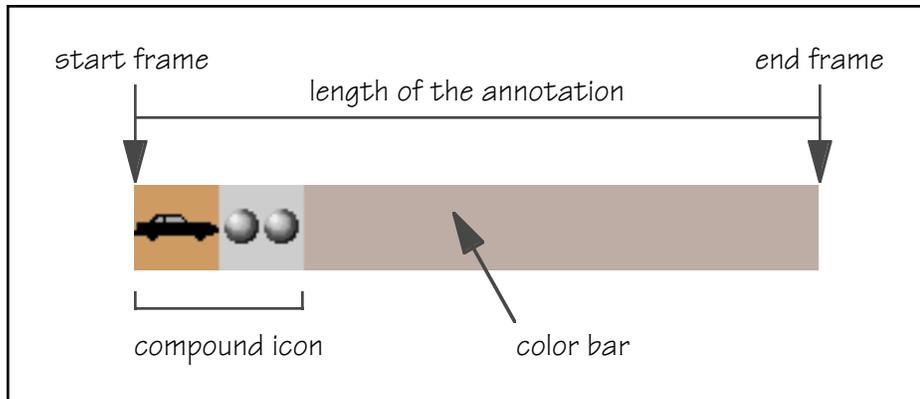


Fig. 107. An annotation is composed of a compound icon and a color bar. This particular annotation refers to the existence of two cars.

Compound Icons

A Compound icon has two principal functions: it **labels** the color-bar of the annotation which follows it, and indicates by its left edge the onset (or **start-frame**) of its annotation's validity. The source of the Compound icons is the Icon Space, where they are constructed and atemporally stored. Compound icons can be **selected** by clicking upon them (indicated by an animated outline), offering a small set of standard Macintosh operations: **cut** (**⌘X**), which deletes the Compound icon and its annotation, and moves it to the Macintosh Clipboard; **copy** (**⌘C**), which copies a Compound icon to the Macintosh Clipboard; and **paste** (**⌘V**), which allows you to paste a previously-cut or previously-copied Compound icon from the Clipboard to the insertion point defined by the Select Bar. Compound icons can be multiply selected by holding down the Shift key as you click on each.

Compound icons can also be dragged to the Select Bar (where they will duplicate themselves into the Compound icons of new annotations); dragged to the Workshop in the Icon Space (where they will open the Workshop hierarchy out to their components), or dragged to the Query Bar in the Icon Space (where they will create a new **filter-unit**, or query). In each of these latter three cases, the Compound icons themselves and their annotations will not be removed from the Media Time Line.

Select-Bar Icons

The content of the current frame is continually displayed on the right side of the Select Bar in a set of icons known as the Select-Bar icons. These icons report the Media Time Line-icons of the annotations beneath the Select Bar, making visible what might be hidden from view while offering a slice through the content of the Movie at the current frame. Thus, the annotations for any given frame of the Movie can be known just by reading down the Select-Bar icons. In

Figure 2, for example, we can see that “it is a day in the 1920’s; the scene takes place in front of a mountain; the weather is clear; there is a car; the car is falling down; it is a medium shot; the camera is tilt-tracking; and the recording medium is black-and-white-film.”

The **Select-Bar** icons have an important use in the editing of annotations: you can **crop** an annotation’s end-frame to the current frame by dragging its **Select-Bar** icon off the Media Time Line to the Macintosh Desktop. **Select-Bar** icons can also be drag-copied to the Workshop in the Icon Space (where they will open the Workshop hierarchy out to their components), or drag-copied to the Query Bar in the Icon Space (where they will create a new **filter-unit**, or query).

Select-Bar icons can be **selected** by clicking upon them (indicated around the icons by an animated outline), offering a small set of standard Macintosh operations: **cut** (⌘X), which crops its annotation, and moves the icon to the Macintosh Clipboard; **copy** (⌘C), which copies a **Select-Bar** icon to the Macintosh Clipboard; and **paste** (⌘V), which allows you to paste a previously-cut or previously-copied **Select-Bar** icon from the Clipboard to the insertion point defined by the **Select Bar**. **Select-Bar** icons can be multiply selected by holding down the Shift key as you click on each; or you can select **all** of the visible **Select-Bar** icons with the key-combination ⌘A. This can be helpful, for example, if there is a cut to a new shot in the Movie over which no annotations of the preceding shot hold true: selecting “all” and cutting the **Select-Bar** icons when the **Select-Bar** is on the first frame of the new shot will terminate all of the prior shot’s annotations. The **Select-bar** icons could then be pasted into the Media Time Line en-masse at a later time, if a shot similar to the first shot appeared.

The Settings Palette



One of Media Streams’ supplementary features is a small floating window called the **Settings Palette**. The **Settings Palette** has four buttons at its top and a rectangular display region beneath them. These four buttons allow you to select the settings of, from left to right, the **Log Mode**, the **Audio Scrub Setting**, the **Annotation Extend Mode**, and the **Animated Icons Setting**. These modes and settings control, respectively: whether the Icon Information Editor is called up in the creation of a Compound icon, and whether a newly compounded icon appears on the Media Time Line; whether the Audio is heard when scrubbing, or not; whether newly-inserted annotations extend to the end of the Movie, or to the next shot boundary; and whether the system’s Animated icons are animating or not. At the bottom of the **Settings Palette** is the **Current Annotator Display**, which displays the “character” icon you used to identify yourself with when you logged in at the start of your session.

The Media Streams Icon Space

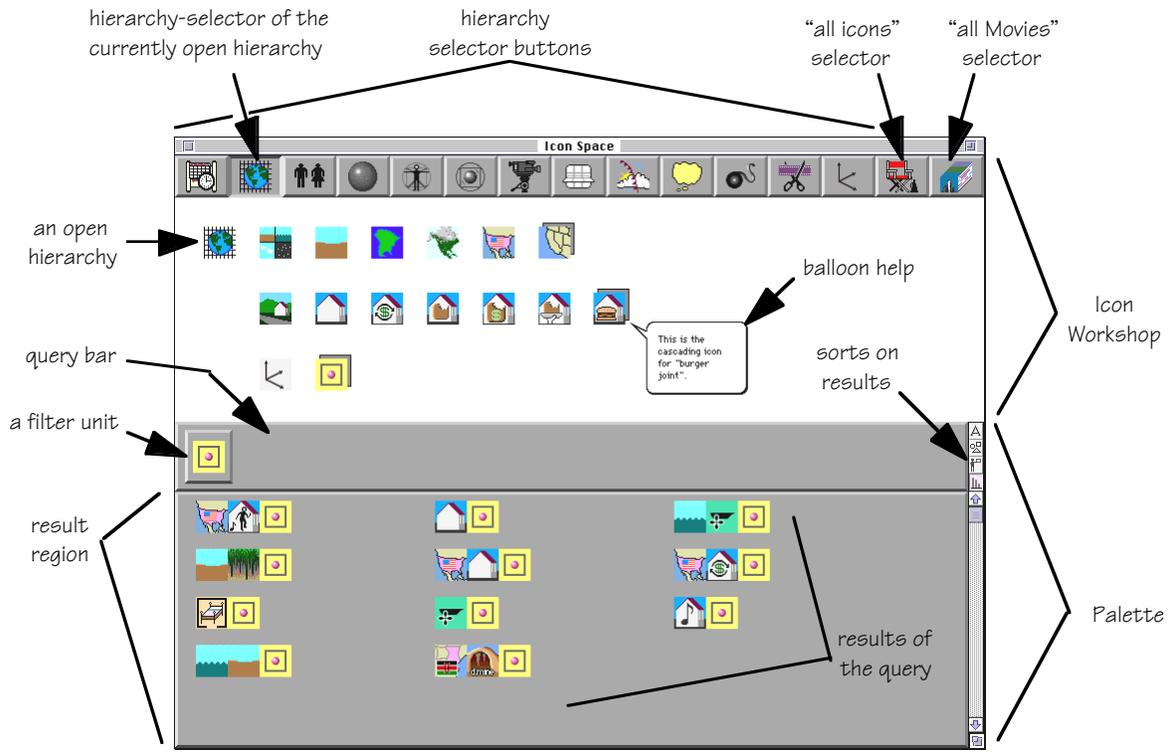


Fig. 108. The Icon Space

The Icon Space

The **Icon Space** is the interface for the selection and compounding of the icons in Media Streams. It is your view onto the atemporally indexed space of iconic descriptors. It displays the **Icon Workshop**, which is a navigable, hierarchically-structured dictionary of the system's iconic descriptor set, and an **Icon Palette** region, where you may store and retrieve the icons used in your (and others') annotations. Together, the Workshop and Icon Palette allow you to quickly create and locate the iconic descriptors you need.

Icons

Icons are the currency of Media Streams: they are ever-present in all Media Streams operations concerned with the annotation, retrieval, and assemblage of digital video and audio. There are **three distinct types** of icons used in

Media Streams: **Workshop icons**, **Compound icons**, and **Media Time Line icons**.

Workshop Icons

The icons found in the dictionary-like Workshop hierarchy are the vocabulary of graphic elements from which all of the other icons in the system — with the exception of **Media Time Line icons** — are constructed. Workshop icons are always **unary** pictographs, readable only as pure nouns, verbs, adjectives, directions, or numbers, etc. This does not mean, however, that all single-element icons are Workshop icons. If a Workshop icon is dragged from the Workshop onto the Media Time Line, for example, it becomes a unary **Compound icon**. Workshop icons are only found in the Icon Space Workshop.

Compound icons

While workshop icons are used to make Compound icons, Compound icons are used to describe the content of the Movie. Compound icons appear on the Media Time Line (where they index states and events in the Movie), and in the Icon Space's Palettes (where they are stored and can be searched for).

Compound icons can contain up to seven constituent Workshop icons. It may be helpful to think of Workshop icons as the “atomic” or “elemental” words of your descriptive vocabulary, while Compound icons are the “molecular” phrases or sentences with which you describe the Movie.

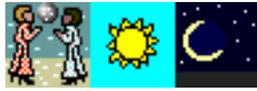
Some confusion may arise over the distinction between those Compound icons which only have one element, and the Workshop icons which are visually identical to them. In fact, there is a great difference, as great as the difference between the word “dog” and the sentence, “There is a dog.” The word “dog” is a part of our vocabulary, whereas the sentence “There is a dog” is a description of the world around us or the Movie in front of us. Try to keep this distinction in mind when you deal with or otherwise verbally translate all Compound icons.

There are several ways to create Compound icons. A very simple way to create an Compound icon, if the Compound icon you want has only one constituent Workshop icon, is simply to drag the desired Workshop icon from the Workshop to the Media Time Line. Creating Compound icons with more than one component requires slightly more complex user actions, and depends on the type of Compound icon you wish to make. There are **three types** of Compound icons — **Ordinary Compound icons**, **Glommed icons**, and **Compound Transition icons** — and each is created in its own way.

- **Ordinary Compound Icons**

The first type of Compound icons is **Ordinary Compound icons**, which contain up to three elements from the **same** Workshop hierarchy.

Because the elements of a Compound icon are all from the same hierarchy, each successive element adds a further specification to the idea of the icon. Examples are:



the **time** icon for “the scene occurs on a summer evening in the 1970’s”



the **space** icon for “the scene is located on top of a street in Texas”



the **character** icon for “there are two adult-female dentists”



the **object** icon for “there are three blue horses”

Notice how each element of these Compound icons belongs to the same hierarchy as the ones to which it is bound, at the same time that it further specifies the idea of the Compound of which it is a part.

Multiple-element Compound icons are assembled in the Icon Workshop by holding down the **⌘** key; single-element Compound icons can be made with the **⌘** key, or created more directly by dragging a Workshop icon to the Media Time Line. Details about constructing all manner of Compound icons are given in the section, “Creating a Compound Icon” (p.387).

- **Glommed Icons**

The second type of Compound icons is **Glommed icons**, which combine up to three Ordinary compound icons across **different** descriptor hierarchies. Glommed icons get their name from the special method, called **glomming**, by which they are created. Unlike Ordinary Compound icons, each successive element of a Glommed icon adds an entirely new dimension of content to the icon. A Glommed icon might contain, for example, some combination of icons from the “characters,” “objects,” “screen position,” “relative position,” “characters’ actions,” or “objects’ actions” hierarchies.

Even though they contain elements from different Workshop hierarchies, Glommed icons are only used to annotate the content of a single stream. Thus, the Glommed icon for “a man pats a dog,” while comprised of a character, a character action, and an object, would only be used to describe the “character action” currently taking place in the Movie. Descriptions of the man, or the dog, would be found in other Media Time Line streams.

Examples of Glommed icons are:



the **objects' screen-position** icon for “airplane in the vertical center of the screen” (glomming together an object and a screen-position)



the **characters' action** icon for “a woman sneezes” (glomming together a character and a character action)



the **characters' action** icon for “a man pats a dog” (glomming together a character, a character action, and an object)



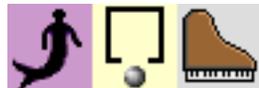
the **object action** icon for “a cloud moves screen right” (glomming together an object, an object action, and a screen position)



the **objects' relative position** icon for “meat is inside an oven” (glomming together an object, a relative position, and an object)



the **characters' action** icon for “Sylvester is eating Tweety” (glomming together a character, a character action, and a character)



the **characters' relative position** icon for “a mermaid is in front of a piano” (glomming together a character, a relative position, and an object)

Glommed icons are assembled on the Media Time Line, and can be found on the Media Time Line and the Icon Space Palettes. Details about constructing all manner of Glommed icons are given in the section, “Creating a Glommed Icon” (p.387).

- **Compound Transition Icons**

The third type of Compound icons are **Compound Transition icons**, which contain one icon from the Transitions hierarchy of the Icon Workshop, and up to two “micons” (iconified Movie segments) derived from scenes in the video. In order to describe the Movie’s cinematographic transitions, Compound Transition icons concatenate an icon of the last frame of the shot before the transition, with a descriptive Transition icon, followed by the first frame of the shot immediately following the transition. Compound Transition icons are automatically assembled in the Transitions stream of the Media Time Line when you drag an iconic element from the Transition

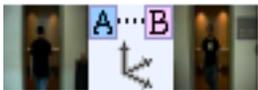
hierarchy to the Media Time Line. Examples of some Compound Transition icons are:



simple cut



forward temporal ellipsis of a determinate length



spatial translation of a determinate proximity

Icon Titles

Compound icons, as specializations or instantiations of Workshop icons, may be **labeled** with a title or number. So, for example, the icon for



“John grabs microphone” contains a labeled instantiation of an adult male,



while an instantiation of a Californian eatery might be labeled “Tofu Hut.”

These textual labels are constructed with the Icon Title Editor facility that can be called up from the Icon Information Editor (see p. 375).

Media Time Line Icons

Media Time Line Icons are three-dimensional extruded icons that represent segments of video. They appear only in the Icon Space's Palettes, where they are shown as the results of queries for Movies, and also where they can be used to further filter a query. More information about the use of Media Time Line Icons can be found in the section, “Some Representative Queries” (p.394).

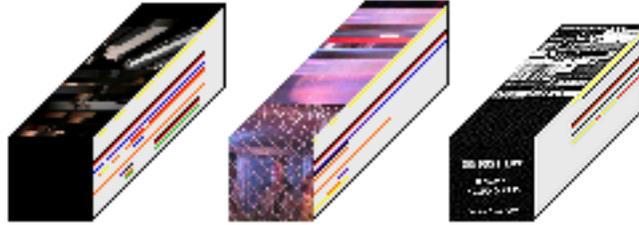


Fig. 109. Three Media Time Line Icons, representing three short Movies or scenes from Movies. It is easy to determine which of the Movies are black-and-white or color from the top edge of their Media Time Line Icons' extruded views. The right-side panels of the Media Time Line Icons give a rough picture of the extent to which their Movies are annotated. The length of a Media Time Line Icon is logarithmically related to the length of the Movie it represents.

The Icon Workshop

The dictionary-like, hierarchically-organized Icon Workshop is an interface to the vocabulary of graphic elements from which all of the system's Ordinary Compound icons are constructed. The Icon **Workshop** occupies the upper half of the Icon Space window and consists of a large white display region, and, above it, a horizontal row of buttons. In the Icon Workshop, cascading icons are organized in **hierarchies** from levels of generality to increasing levels of specificity. The row of **Workshop buttons** at the top of the Icon Space window permits access to the thirteen hierarchies: clicking on one of these buttons* opens up its hierarchy in the Workshop's white display region. There are thirteen top-level hierarchies, or classes of information, into which all of the Workshop's iconic descriptors are divided: Time, Space, Characters, Objects, Character Actions, Object Actions, Cinematography, Screen Position, Weather, Thoughts, Recording Medium, Transitions, and Relative Position. Broadly speaking, these hierarchies correspond to annotation-streams in the Media Time Line. Each of these hierarchies are discussed at length in the next section, "Descriptor Hierarchies" (p.372).

* With the exception of the last two buttons, to whose functionality we shall return later.



Fig. 110. The Icon Workshop

Observe Figure 110 on this page. At the top of the Icon Workshop is the row of Workshop buttons. Each of these buttons corresponds to a hierarchy of icons and “opens up” its corresponding hierarchy in the white Workshop space when it is selected. Reading from left to right, we see the Workshop buttons for Time, Space, Characters, Objects, Character Actions, Object Actions, Cinematography, Screen Position, Weather, Thoughts, Recording Medium, Transitions, and Relative Position. The Workshop button for the Space hierarchy is inverted, indicating that it is selected; we can see that the Space hierarchy, consequently, is expanded beneath it. The last two Workshop buttons on the far right are special buttons which stand for, respectively, the set of all icons and the set of all Movies; the purpose of these Workshop buttons is discussed later, in the section “Operations with the Icon Palette” (p. 392).

Below the row of Workshop buttons is the Workshop’s display region, where icon hierarchies are opened out. Once opened, the top-level icon hierarchies divide **horizontally** into “exclusive” categories and **vertically** into “inclusive,” compoundable ones. Exclusive descriptive categories are those which **cannot** be true at the same time, such as a location in both America and France, or a moment in time which occurs simultaneously in the 1970’s and World War II. Inclusive, compoundable categories, on the other hand, are those which **can** be true at the same time: nothing prevents a location in America from also being inside a hotel, or a moment in time from occurring in the 1970’s, and also on a summer evening. Each icon hierarchy branches into no more than three vertical (compoundable) categories. Compound icons may hence contain up to three elements, each element drawn from a different compoundable category.

In Figure 110, the Space hierarchy has been opened out, and a path through the hierarchy has been navigated. Space is divided **vertically** into geographic space, functional space, and topographical space; in order to read each of its three vertically-divided, “inclusive” hierarchies, begin at the left and read to the right. This user has wended a path through Geographic space

which begins at “land” and moves through “continent” to “North America” and the “United States of America,” terminating at the “South-Western States.” The functional space sub-hierarchy begins with “building space,” moving then from “commercial building” to “food-related building” to “food-selling facility” to “restaurant,” and ending with “burger joint.” Topographical space has barely been opened at all: this user has only navigated as far as the general descriptor for “inside.” Notice how the three rightmost icons — “South-Western States,” “burger joint,” and “inside” — are shadowed by gray squares, indicating that they are not the terminal nodes of their hierarchy. Each has more icons beneath it.

The primary method of navigating through the hierarchies is by clicking on the icons themselves. A **gray shadow square** shaded by a Workshop icon indicates that that icon is the superordinate of more specialized subordinate icons; clicking on that shaded icon opens up its sub-hierarchy of subordinate icons. Several or all of those subordinates may also be shaded, and so forth. An unshaded icon, on the other hand, is a terminal node in the icon hierarchy and possesses no subordinates. Terminal nodes may appear anywhere in a hierarchy, and may have siblings with many, many descendants. More details about navigating the Workshop are described later in “Navigating the Workshop” (p. 385) and “Creating a Compound Icon” (p. 387).

Icons that you have passed on your path to their daughters remain visible, so that an opened hierarchy can be “read” from left to right. This “reading” marks a trace through increasing specificity: for example, one might read the path made to San Francisco: “Land – Continent – North America – United States – South-Western States – California – San Francisco.” Branches in the cascading hierarchies can be closed by clicking on parents higher in the tree; double-clicking on upper-level parents will re-open their sub-hierarchies.

Take some time to familiarize yourself with the Workshop’s iconic hierarchies. Balloon help, which can be turned on or off with the “help” key, will display small balloons with icons’ titles when the cursor is over an icon. The thirteen hierarchies of the Icon Workshop are described below.

Descriptor Hierarchies



Time

“Time” is **when** a shot appears to take place, such as the a summer evening in the Middle Ages, the 1970's, the Future, or World War II. Time is **vertically** subdivided into historical period (from the age of the dinosaurs through the twentieth century on into the future), time of year (spring, summer, fall, and winter), and time of day or night (morning, afternoon, sunset, midnight, etc.).



Space

“Space” is the setting or location **where** a shot appears to take place, such as a beach, an island off the coast of Australia, inside a submarine, or on top of a factory in the Mid-Western United States. Space is vertically subdivided into geographical space (land, sea, air, and outer space), functional space (buildings, public outdoor spaces, wilderness, and vehicles), and topological space (inside, outside, above, behind, underneath, etc.).



Characters

“Characters” refers to **who** appears in a shot, such as the president, a policewoman, the ghost of a football player, three young McDonald's employees, a large crowd of hippies, a space-alien, or Homer Simpson. Characters are **vertically** subdivided into characters (female, male, unknown gender, non-human, and crowd), occupations (personal care, commercial, institutional, religious, sports) and number (one, two, three...many).



Objects

“Objects” refers to **what** objects appear in a shot, such as a vacuum cleaner, a cigarette, a fence, a cow, or a lemon. Objects are **vertically** subdivided into types of objects and number (one, two, three...many) of objects.



Character actions

“Character Actions” are the actions that characters perform in a shot, such as running, nodding, blinking, or shaking hands. Character actions are **horizontally** subdivided into actions involving a single character, two characters, or groups of characters. Each of these are subdivided into full body actions, head actions, arm actions, and leg actions; these are further divided between conventionalized physical motions and abstract physical motions.



Object actions

“Object Actions” are the actions that objects perform in a shot, such as falling, sliding, melting, colliding, or exploding. Object actions

are **horizontally** subdivided into actions involving a single object, two objects, or groups of objects. Each of these is divided according to object **motions** and object **state changes**. For example, the action of a ball rolling is an object motion; the action of a ball burning is an object state change.



Cinematography

“Cinematography” icons describe the states and actions of the camera recording a shot, such as pans, zooms, tilts, canting, tracking, and trucking. Cinematography is **horizontally** subdivided into lens actions (framing, focus, exposure) tripod actions (angle, canting, motion), and truck actions (height and motion). By layering these iconic descriptors on the Media Time Line, simple to very complex camera motions can be described.



Screen position

“Screen position” refers to the position on the screen of a character or object, or the direction in which a character or object performs an action, used for such expressions as “Pat is on the left side of the screen” or “Chris is walking toward screen right.” Screen positions are **horizontally** subdivided into two-dimensional screen position and screen depth.



Weather

“Weather” is the apparent weather of a shot, such as rainy and windy, sunny, or snowing and calm. Weather is **vertically** subdivided into moisture (clear, partly sunny, partly cloudy, overcast, rainy, and snowy) and wind (no wind, slight wind, moderate wind, and heavy wind). Temperature is not something that can be directly seen; a video of a cold clear day may look exactly like a video of a hot clear day. It is the presence of snow or ice that indirectly indicates the temperature.



Thoughts

“Thoughts” refers to the annotator’s subjective thoughts about the shot, such as “too busy,” “drab,” “colorful,” or “good.” Thoughts are **vertically** subdivided into thoughts about the screen (framing,

activity, color) and evaluation (from three thumbs up to three thumbs down).



Recording Medium

“Recording Medium” refers to the type of media onto which the shot appears to have been originally recorded, such as grainy black-and-white 35-mm film, 1-inch color video, or sepia-toned 8-mm film. The Recording medium context is **vertically** subdivided into stock (70 mm film, 8mm video, etc.), color quality (color, black & white, sepia, etc.), and graininess (fine, medium, coarse, etc.).



Transitions

“Transitions” refers to the type of transition used between two shots, whether spatial, temporal, or visual. Some examples are a cut, a dissolve, a wipe, a match-cut, or a flashback. Transitions between shots are **horizontally** subdivided according to temporal transitions (e.g., continuous, forward ellipses in time of a determinate length, forward ellipses of an indeterminate length, and the corresponding transitions in which there is a temporal reversal), spatial transitions (e.g., continuous transitions in which spatial proximity is determinate, and transitions in which spatial proximity is indeterminate), and visual transitions (cuts, wipes, dissolves, etc.).



Relative position

“Relative Positions” are the relative position of one character or object in relation to another character or object, used for such expressions as “Pat is in front of Chris,” “Jean is on the threshold of a building,” or “the apple is on top of a table.” Relative positions are **horizontally** subdivided into: inside, on the threshold of, outside, on top of, underneath, above, and below.

The Icon Information Editor

The Icon Information Editor is a special facility for augmenting the content of an iconic descriptor. It is called up whenever the Log Mode is set to “Compound with Editing,” and you have command-clicked to create a Compound icon in the Workshop.

The editor consists of a text field flanked by a column of buttons. The text field contains an automatically-generated transcription of the icon's content, and allows you to add more textual information, such as facts or opinions, about the icon's subject.

The column of buttons flanking the Icon Information Editor's text field call up a number of supplementary dialogs for entering information. These include facilities for establishing or editing an icon's textual title or label; the color of its subject matter (e.g., declaring that a car is "yellow"); or, for "Time" icons, the precise date and time the icon represents.

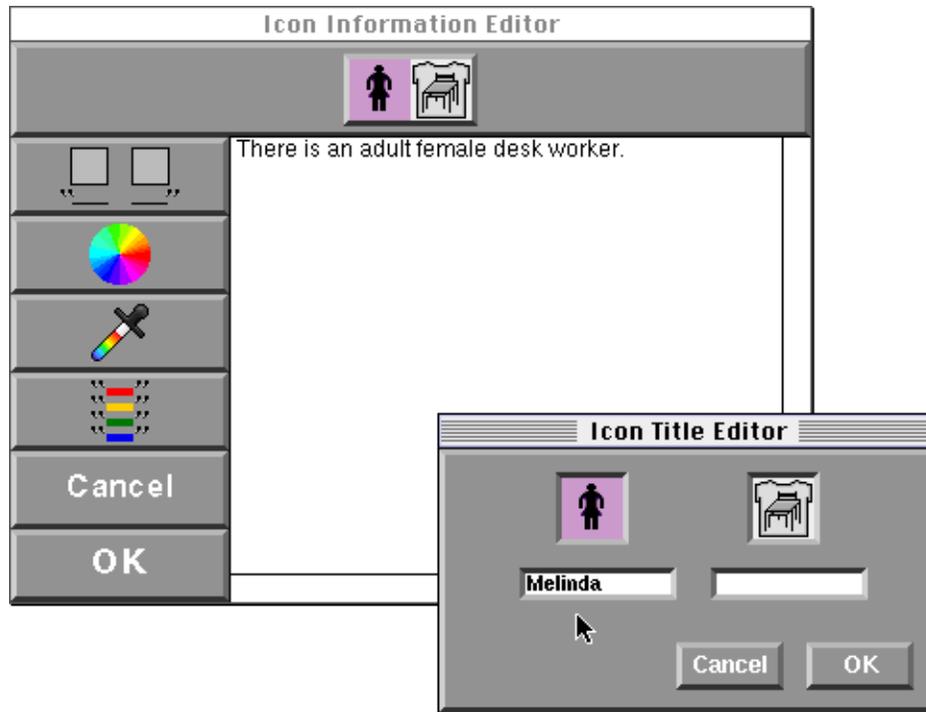


Fig. 111. The Icon Information Editor and the Icon Title Editor. The Icon Title Editor, which permits you to add a title or number to the Compound icon, is called up when you click on the topmost button on the left-hand side of the Icon Information Editor.

The Animated-Icon Editor

The Animated Icon Editor allows you to re-organize the frames that comprise an Animated icon, or create a new Animated icon by modifying an old one. You can open more than one Animated-Icon Editor at a time and drag icon-frames between them, making it possible to mix and match parts of animated icons in the construction of new ones. The Animated Icon Editor also allows you to change the speed of an animated icon. You can call up the Animated Icon Editor by control-clicking on the animated icon you wish to edit.

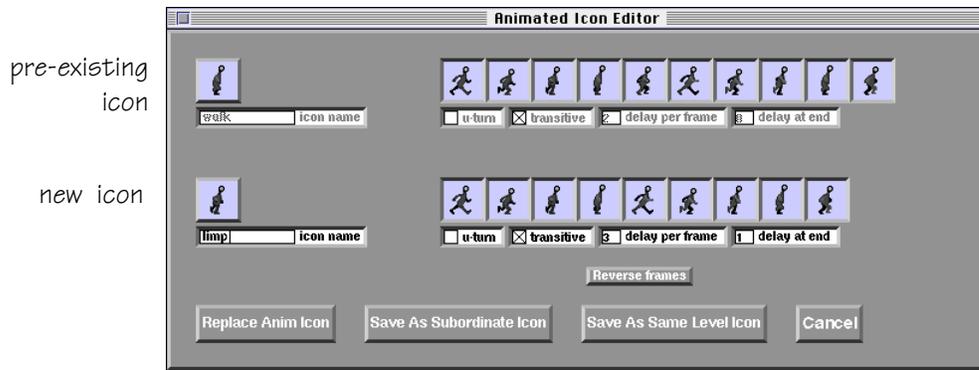


Fig. 112. The Animated Icon Editor. In this example, the user has created an icon for the action of "limping" from a subset of the icon-frames for "walking."

The Icon Palettes

The Icon Palettes are your interface for retrieving previously-defined Compound icons. They are broken up into two main components: the Query Bar, where you make queries into the space of defined icons, and the Result-region, where the results of your query are displayed. You can have more than one Icon Palette open at a time, with different queries on each.

At the top of an Icon Palette is the **Query Bar**. Queries appear as Compound icons inside small rectangular regions called **filter units**. Filter units represent individual queries, and it is possible to have many filter units in the Query Bar at the same time, returning many query-results all sharing the space of the Result-region. The Media Streams **query language** is a set of graphic conventions which allow you to combine filter units in highly complex ways, creating highly specific queries. By using Movie-icons in your query, you can even make queries for other Movies or parts of Movies that satisfy the conditions you specify; the results returned by such queries are displayed as Media Time Line Icons. More information about making queries and using the query language can be found in the section, "The Query Language" (p. 393).

The large lower section of an Icon Palette is the **Result-region**, in which Compound icons satisfying the terms of the query are displayed. If there are several simultaneous queries, the results will be "paragraphed" in the Result-region, separated into distinct zones by horizontal divisions. The Icon Palette Result-region has a vertical scroll-bar which allows you to see results that have extended out of the Icon Palette's range of view.

At the upper right of the Icon Palettes, on the right side of the Query Bar, are the **Result-Sorting Buttons**. These allow you to sort the results of your queries by a number of methods: alphabetically, by annotator, by icon type, and by frequency of use. If no sort is specified, the Compound icons in the Result-region will be sorted by their recency of use.



Fig. 113. An icon Palette residing in its own window. Across the top is the Query Bar, in which rectangular filter units are made in order to produce queries into the space of icons. The large gray rectangle below is the Result region, in which Compound icons satisfying the terms of the query are displayed. In this example, a query has been made for those Compound icons which simultaneously contain a screen position **and** an object. Note how some of the results use a screen position icon to describe an object's screen position, while others use screen position icons to illustrate the direction of an object's action.

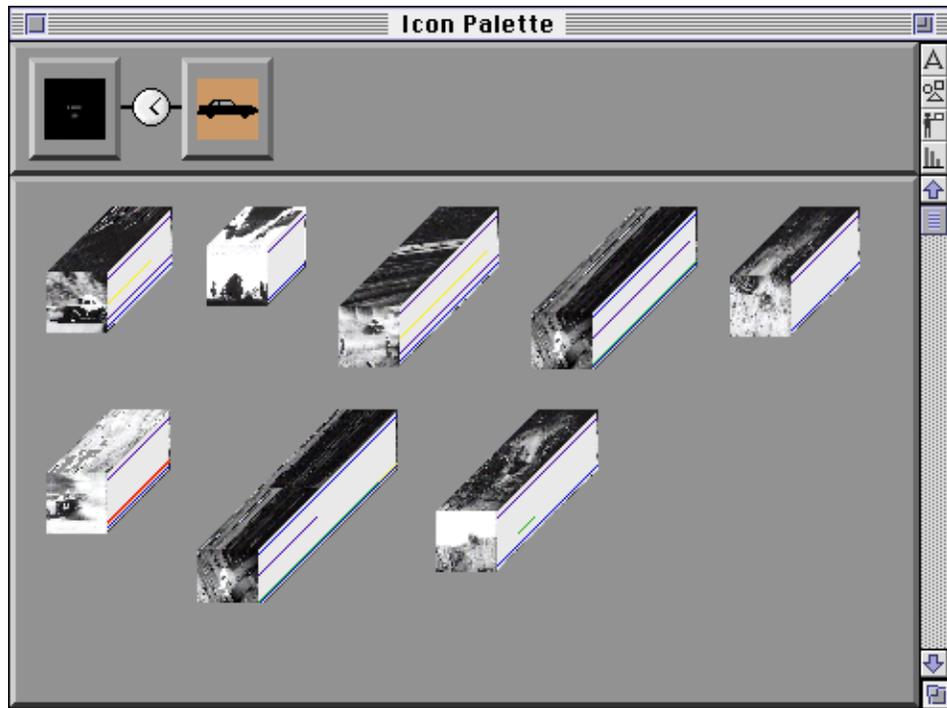
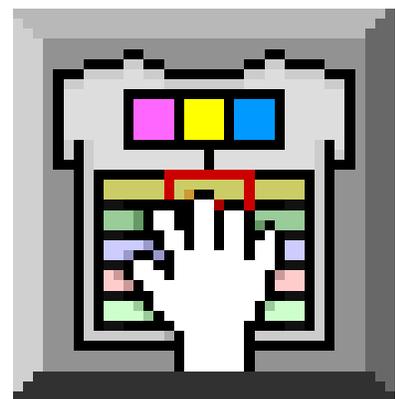


Fig. 114. Some queries return Media Time Line Icons. The query shown returns all of the segments from Bruce Connor's film **A Movie** that contain cars. If the filter units in the Query Bar were reversed, the query would return all of the "car" icons that were used to annotate Bruce Connor's **A Movie**.

Part Two



Using the System

Getting Around the Movie

There are four interface-tools for moving around and navigating the video: the Movie Controls, the Minutes and Seconds Thumbnails, the Minutes and Seconds Scrubbers, and the Select Bar.

Getting Around the Movie with the Movie Controls

- | | | |
|---|---------------------------|--|
|  | Pause | Pauses the movie if it is playing, playing an extent, or scanning. |
|  | Play | Plays the Movie from the current frame. |
|  | Reverse Play | Plays the movie backwards from the current frame. |
|  | Play Extent | Plays that segment of the movie whose start-frame and end-frame are determined by an annotation or pair of shot boundaries, and which contains the current frame. If no stream is selected, this control will play the current scene (defined by the shot boundaries immediately bracketing the current frame — often discernible in the Videogram, and represented elsewhere in the Transitions stream). If a stream has been selected, this control will play the segment of the Movie which corresponds to the annotation in the selected stream and contains the current frame. If there is no annotation in the selected stream at the current frame, nothing will be played. If the current frame has multiple annotations that describe it (as could happen if multiple streams were selected or if the selected stream was expandable), the "Play Extent" control will play from the beginning of the first current annotation to the next change in the selected stream(s). |
|  | Reverse Play Extent | Reverse-plays the segment of the movie containing the current frame and whose endpoints are defined by an annotation in a selected stream or the shot boundaries bracketing the current frame. |
|  | Frame Forward | Advances the current frame by one. |
|  | Frame Reverse | Diminishes the current frame by one. |

	Jump Forward	(Jumping by content) Jumps the current frame to the next change in the selected stream(s). If no stream has been selected, this control will advance the current frame to the next shot boundary.
	Jump Reverse	Jumps the current frame to the previous change in the selected stream(s). If no stream has been selected, this control will jump the current frame to the prior shot boundary.
	Scan Forward	Plays the movie at ten times its normal play speed.
	Scan Reverse	Plays the movie in reverse at ten times its normal play speed.

Content Navigation with the Movie Controls

Several of the Movie Controls allow you to jump around the Movie by changes in its content. In order to do so, select an annotation stream (by clicking on and inverting its Stream Control) whose content is relevant to your navigation. For example, if you wished to advance the Movie to the next character-change, you would select the Characters stream. The Jump Forward and Jump Reverse buttons in the Movie Control bar will then jump the current frame to the next or previous change in the Characters stream. Clicking on either of the Play Extent controls will, instead of **skipping** through the Movie, play segments of the Movie described by annotations in the selected stream. If no streams are selected, these content-navigation controls will use the Movie's shot boundaries as a default.

Getting Around the Movie with the Thumbnails

The Thumbnail icons in the Minutes and Seconds Movie Streams offer a fast but coarse way of skipping through the Movie. Thumbnails will highlight in blue when they are clicked upon, indicating precisely which Thumbnail your cursor is positioned over; double-clicking on a Thumbnail will jump the current frame to the frame represented by the Thumbnail. Because the Minutes Thumbnails stream subsamples the Movie at the rate of one frame per minute, and the Seconds Thumbnails at the rate of one frame per second, however, moving to a point indexed **between** two Thumbnails may require finer adjustments. These adjustments can be made with the Minutes and Seconds Scrubbers and the Select Bar.

Getting Around the Movie with the Scrubbers

Two red, rectangular Scrubbers reside in the Minutes and Seconds Movie Streams. Clicking on and dragging these Scrubbers will allow you to move through the Movie at scales 225 and 3.75 times greater than the scale of the

Select Bar and Annotation Streams. As you drag one of the Scrubbers, the Movie window displays what the new current frame will be when you release the Scrubber. The Seconds Scrubber is particularly useful for bringing nearby but obscured portions of the Movie into annotatable range.

Getting Around the Movie with the Select Bar

For the purpose of navigating the Movie, the Select Bar can be thought of as a Frame Scrubber. It is the scrubber with the greatest precision but the smallest working range, as its utility for editing is limited to the 7.73 seconds of Movie-time represented in the Videogram. In fact, the Select Bar does allow you to scrub past the edges of the Media Time Line window; but you will not be able to edit annotations with the Select Bar when it is out of view. If you let go of the Select Bar while it is off-screen, however, it will pop the current frame to an editable location on the Media Time Line.

To use the Select Bar, bring your cursor over the Select Bar until it becomes an open  hand; depressing the mouse button will close the hand , allowing you to drag the Select Bar horizontally while scrubbing through the Movie. You can also “pop” the Select Bar to any visible point in the Media Time Line by clicking on one of the frame-like dividers that separate the streams; the Select Bar will jump to the horizontal location of the point you clicked, and the current frame will jump to the frame indexed by the Select Bar's new location.

Operations With Streams

Moving Streams Around

You can dynamically re-arrange the streams on the Media Time Line as you see fit. Clicking on and dragging a terminal Stream Control (one of the Stream Controls furthest to the right) will allow you to move the vertical position of its stream within the stream's group; clicking on and dragging a superordinate Stream Control (one of the Stream Controls to the left of the terminal controls) will allow you to vertically slide its entire group of streams around.

Selecting and De-Selecting a Stream

To select a stream, single-click on its Stream Control and release the mouse button; its beveled Stream Control will invert, indicating your selection. To select multiple streams, hold down the “Shift” key while single-clicking on the Stream Controls of the ones you want. To de-select a selected stream, single-click on its inverted Stream Control; the Control will revert to its usual state.

Selecting an hierarchically non-terminal (i.e., superordinate) Stream Control has the same effect as selecting all of the streams under it.

Hiding a Stream

To hide a stream, double-click on its Stream Control. The stream will disappear; the Media Time Line will then automatically compact itself, and the stream's control-icon will appear on the correct Hidebar. Double-clicking on a superordinate Stream Control will hide all of its subordinate streams under its control-icon.

Recalling a Stream from the Hidebar

To recall a stream that has been hidden, double-click the icon which represents it on the Hidebar. Double-clicking on the icon of a superordinate grouping of streams will open all of its streams that were not explicitly hidden before. If you cannot find the icon of your desired stream on the Hidebar (and the stream is not already visible on the Media Time Line), it will be necessary to recall the superordinate group which contains the stream you want; double-click on the superordinate group's icon on the Hidebar. The Media Time Line will automatically hide some streams if you recall more than it has room to display. Holding the **⌘** key while you double-click on a Hidebar icon will open **all** of its subordinate streams.

Navigating Through the Movie by Content

Select a stream or group of streams according to whose content you wish to browse the Movie. Clicking on the Jump Forward or Jump Reverse Movie Controls will skip the current frame to the one indexed by the next change in your selected streams; clicking on the Play Extent or Play Reverse Extent Movie Controls will play the Movie segment indexed by the annotation in which the current frame is embedded. If multiple streams are selected, or if the selected stream is expandable, a logical “or” will be run across the streams in the search for content-changes.

Saving Your Annotated Media Time Line

To save a Movie's annotated Media Time Line, you can: Click on the Media Time Line you wish to save, making it the current window; and then select “Save” or “Save As” in the File Menu to save the Media Time Line in the standard way. Alternatively, you can go directly to the Media Streams Menu and select “Save Media Time Line As.” As a shortcut, you can use the standard **⌘S** key combination to save your Media Time Line. If you do not specify the title of your Media Time Line with the “Save As...” command, it will be given the title “Media Time Line” when you save it.

Operations with the Workshop

Navigating the Workshop

Navigating the Workshop begins with opening up one of the thirteen icon hierarchies into the Workshop's white display region. To do so, click on the Workshop button at the top of the Icon space which corresponds to the hierarchy you wish to explore.

Once opened, the top-level icon hierarchies divide **horizontally** into "exclusive" categories and **vertically** into "inclusive," compoundable ones. A gray square behind a Workshop icon indicates that that Workshop icon is the parent of a sub-hierarchy with children-icons hidden underneath it. Double-clicking on a shadowed Workshop icon will open up its sub-hierarchy (and hide its gray shadow square). Icons that you have passed on your path to their daughters remain visible, so that an opened hierarchy can be "read" from left to right. Branches in the cascading hierarchies can be closed by clicking on parents higher in the tree; double-clicking on upper-level parents will re-open their sub-hierarchies. If the category you want is not visible, and you know that it exists in the hierarchy you currently have open, click twice on your desired category's superordinate category. To close an opened branch of a Workshop hierarchy, click once on the icon which represents the category.

Bubble Help

Bubble Help allows you to see the names which identify icons. Pressing the "Help" key, or turning on Bubble Help in the Menu Bar, will enable small text-bubbles to appear when you pass over a Workshop or Compound icon. This feature can be useful when you are learning how to read the Workshop hierarchies, or when the meaning of an icon is unclear.

Bubble Help can also help you "read" a path you have navigated through the Icon Workshop, to ease your learning of the icon hierarchies. Bubble help also provides some information about the other system components.

Filtering the Workshop with an Ordinary Compound icon

You can quickly navigate the Icon Workshop by dragging an Ordinary Compound icon into it from the Media Time Line or Icon Palette. This feature can be helpful if you wish to construct an icon similar to or sharing components with a pre-existing one, if you wish to retrieve elements of a Compound icon for the purposes of a Query, or simply in order to see what Workshop path was taken to make a certain icon. If you drag a Glommed Icon into the Workshop, Media Streams will only open out the hierarchy of the Glom's first element.

Operations with Icons

The Log Modes: What they Mean

When you construct a new Ordinary Compound Icon, the behavior according to which it is compounded is determined by one of the four **Compounding Modes**. You can select the Compounding Mode you wish to use by clicking on the second button from the left on the Settings Palette; doing so toggles from one state to the next. Only one of the Compounding Modes is operative at any given time.

No matter which Compounding Mode setting you choose, new Compound icons are always entered into the space of Compound icons when they are created. The Compounding Mode allows you to choose whether the icons will additionally be sent to the Media Time Line, and whether or not you wish to be interrupted by the Icon Information Editor.



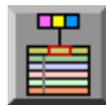
The **Compound-to-Icon Space with Editing** Compounding Mode opens up the Icon Information Editor before sending the new Compound into the space of all Compound icons. This mode allows you to add information to a new Compound icon before it is sent to the Icon Space. You might use this mode if you were preparing all of the Compound icons you would later use to describe a certain Movie, or if you were titling your new icons.



The **Compound-to-Icon Space** Compounding Mode does not open up the Icon Information Editor; instead, the new Compound disappears immediately into the space of all Compound icons, where it will remain unseen until retrieved. You might use this mode if you were preparing all of the Compound icons you would later use to annotate a certain Movie, but do not wish to specify any more than what your icons already describe.



The **Compound-to-Time Line with Editing** Mode opens up the Icon Information Editor before placing the new Compound on the Media Time Line (where it forms a new annotation). The Compound is also sent into the space of Compound icons. This Mode allows you to add information to the Compound before it is sent to the Media Time Line. You might use this mode if you prefer to create new Compound icons as you need them, while you are annotating.



The **Compound-to-Time Line** Compounding Mode does not open up the Icon Information Editor; instead, the new Compound goes directly onto the Media Time Line, where it forms a new annotation. The Compound is also sent into the space of Compound icons. You might use this mode if you prefer to create new Compound icons as you need them, but do not wish to specify any more than what your icons already describe.

A fifth mode of constructing Compound icons, unrepresented on the Settings Palette, is by directly dragging the new Compound, as you create it, from the Workshop to the Media Time Line or Icon Palette. Creating Compound icons in this way will override the current Compounding Mode. As with the other Compounding Modes, the new Compounds you create by dragging will additionally be sent into the space of Compound icons.

Creating a Compound Icon

To create an Ordinary Compound icon in the Icon Space Workshop, navigate through a descriptor hierarchy until the icons you wish to comprise your Compound icon are visible. If the Compound icon you wish to create has only one iconic element (e.g., “unary”), you can drag the element directly from the Workshop to the Media Time Line: a Compound icon will be created using the icon you dragged, and a new annotation will be inserted on the Media Time Line with your Compound at its head. If you wish to create a unary Compound icon, but don't wish to create an annotation with it immediately, you can follow the same instructions below as for multi-element Compounds, or drag it directly to the Icon Palette region.

To create an Ordinary Compound icon comprised of more than one element, navigate through a descriptor hierarchy until the iconic elements you wish to comprise your Compound icon are visible. (Completely unopened vertical divisions of a hierarchy will not contribute a component to the new Compound icon.) Continue until all of the elements you want are the rightmost icons of their sub-hierarchies, and follow immediately after their superordinate parent icon, with none of their siblings visible.

While holding the  key, move your cursor over any of the desired elements; a special  cursor will appear indicating that Media Streams is prepared to compound an icon. Now, depending on the Compounding Mode setting in the Configuration Palette, clicking on one of the rightmost icons will either assemble the compound icon immediately or bring up the Icon Information Editor before proceeding with the assembly.

One useful Compounding shortcut is that not **all** of the Workshop elements you want in a Compound need to be isolated from their siblings at the rightmost end of their sub-hierarchy. With the other elements so specified, you can select the remaining element of a compound directly from an underspecified branch, right out from amongst its siblings.

Creating a Glommed Icon

Because Glommed icons combine up to three different Compound icons across up to three different hierarchies, creating a Glommed Icon requires more complex, and also more precise, user actions. Most Glommed icons are initiated on the Media Time Line by dragging a Screen Position icon, Relative Position icon, Character Action icon, or Object Action onto the Select Bar Icon

of a Character or Object. A Glommed icon containing the Character or Object and the action or position icon will then appear in the appropriate stream.

Glommed icons use an SVO (subject-verb-object) grammar to express their content. The above method describes how to create a Glommed icon with the first two of these pieces in place. Many actions and position descriptions require nothing else — it is sufficient, for example, to say that “René is sneezing,” or that “the cat is in the center of the screen.” Many other actions, however, require a grammatical object to complete their meaning. Take, for example, actions like “Chris kissed **Pat**,” or “Jean walked **to the right**.” In order to add a grammatical object to a Glommed Select-Bar icon, drag the complete icon directly onto the incomplete Glom. The third element will attach itself to the Glommed icon's right edge.

Certain cinematography streams create special Glommed icons. These are: Framing, Focus, and Camera-Look-Through. Each of these takes a Character or an Object as their completion. In addition, the Cinematography icons for Tracking — tripod tracking, canting tracking, tilt tracking, and truck tracking — are a special case of icons that will accept grammatical objects to Glom with, even though the other Compound icons in their stream do not.

Creating a Compound Transition Icon

Compound Transition icons are assembled automatically on the Media Time Line when you drag an Transition icon into the Media Time Line. Thus, you can create a Compound Transition icon by dragging a Transition icon from the Icon Workshop, from the Icon Palette, or even by copy-pasting a Transition icon from elsewhere in the Media Time Line. Media Streams will automatically generate the micons which flank the Transition icon (representing the scenes before and after the transition) and insert the Transition icon at the head of a new annotation. Transition icons representing “cuts,” when applied to the Media Time Line, are (by default) given annotations of a single frame in length.

Viewing, Declaring or Changing an Icon's Information

You can see, declare, or change the information supplementary to any Compound icon by double-clicking on it, bringing up the **Icon Information Editor**. The Icon Information Editor's buttons bring up smaller sub-editors which allow you to edit specific fields of the icon's information. The topmost button, labeled “**Edit Icon Title**,” for example, allows you to individually add a textual title or name to each component of a Compound or Glommed icon. You can use this, for example, to declare that a certain actor is “Sean Connery,” or that a certain location is “Pat's kitchen.” If your title has more characters than can fit across the bottom of an iconic element, Media Streams will prompt you to select an abbreviation with which the icon's title will be viewed; no information is lost when you use such abbreviations.

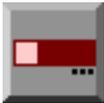
Another feature of the Icon Information Editor is that it allows you to specify the **color** of an Compound icon's subject matter. You can edit an icon's color with any of the three color pickers Media Streams provides: the **Apple Color Picker**, the **Color Name chooser**, and the **Color Dropper**. Clicking on the Apple Color Picker from the Icon Information Editor brings up a dialogue that allows you to choose a color from a standard color wheel; the Color Name chooser allows you to select a color from a short list of common colors; and the Color Dropper allows you to actually sample an object's color by clicking on its image with a  dropper cursor in the Movie window. Many iconic hierarchies, such as screen position and cinematography, don't accept a "color" designation because it would be illogical to do so.

The Icon Information Editor called up when you double-click on a "Time" Compound icon replaces the Edit Icon Title function with "Edit Icon Time." This special editor allows you to quickly and precisely specify the exact date of a Time icon.

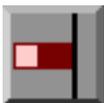
Operations with Annotations

The Log Modes: What They Mean

When a new annotation is created, the heuristic used to determine its default length is specified by one of the two **Log Modes**. You can select the Log Mode you wish to use by clicking on the third button from the left on the Settings Palette. Only one of the Log Modes is operative at any given time.



The **Extend to Movie End** Log Mode sets the default end-frame of all newly-inserted annotations to the end of the Movie. This is a useful setting if you know that the annotations you are about to create are valid for the entire movie (such as "the scene occurs in the 1970's), or over many shots.



The **Extend to Next Scene Break** Log Mode sets the default end-frame of all newly-inserted annotations to the nearest shot boundary immediately after the annotation's start-frame. This Log Mode takes advantage of the shot boundaries calculated by the Preprocessor, and enables you to quickly create annotations which you know to be valid for entire shots or parts of shots.

Making an Annotation on the Media Time Line

To make an annotation on the Media Time Line, the *Select Bar* must be positioned at the point in the Media Time Line at which you want the annotation to begin. To arrange this, scrub through the Movie with the *Select Bar* or *Scrubbers* until the current frame (displayed in the Movie window) is the first frame for which your desired annotation would be valid.

You can now drag the icon you want onto the Media Time Line. Your source icon can come from the *Icon Palette result-region*; from the *Query Bar*; or directly from the *Icon Workshop*, where you just created it. Your source icon can even be a compound icon from an earlier location in the stream you are annotating — but if it is, you should take care not to drag it out of its stream, or you will accidentally remove its annotation!

Another way of inserting a new icon at the *Select Bar* is to *Paste* (⌘V) at the *Select Bar* a Compound icon you have previously *Cut* or *Copied* into the *Macintosh Clipboard*.

As soon as you drag or paste the Compound icon into the Media Time Line, a new annotation will appear at the *Select Bar*. If the icon's intended stream is expandable, the stream will expand upon insertion; if the intended stream is non-expandable, the new annotation will terminate (at the *Select Bar*) the stream's previous annotation.

When there is no possible confusion about where your annotation is supposed to go — for example, a cinematographic “framing” icon cannot appear in any other stream but cinematographic “framing” — the new annotation will be intelligently inserted at the *Select Bar*. This means that, most of the time, you can be considerably sloppy about where you drop the icon you have dragged in. As long as you drop “the scene is located inside a restaurant in California” somewhere on the Media Time Line, *Media Streams* will understand that it is intended for the “Space” stream. There will generally be little confusion when you are annotating with Compound icons.

More precise actions are required of you if there is possible confusion about where your new icon should go. This is almost always the case when you make a *Glommed* icon, and there is more than one possible subject on the Media Time Line that the *Glom* could attach to. If there were two simultaneous “character” annotations on the Media Time Line, for instance, and you dragged a “character action” icon into the Media Time Line, the system wouldn't know which of the characters to *glom* the action with: Is one character performing the action, or the other?

In the absence of further specification, *Media Streams* will solve this confusion by assigning the *glom* to the topmost subject in the topmost applicable stream. Thus, if “Chris” were the upper character annotation in the above example, dragging in the character action “walking” to an arbitrary place in the Media Time Line would insert a “Chris walking” annotation at the *Select Bar*. This is a default behavior; in order to precisely specify which of the

possible subjects the new icon should glom to, you must drag the icon directly onto the subject you wish to annotate. Thus, if “Pat” were the character with the lower annotation, and you wished to express “Pat walking,” you would have to drag the “walking” action directly onto Pat’s Media Time Line or Select-Bar icons. You will need to take this care when annotating any of the Glom-containing streams, such as “Character Actions,” “Object Actions,” “Character Screen Position,” “Object Screen Position,” “Character Relative Position,” “Object Relative Position,” “Camera Relative Position,” and certain cinematography annotations. For more information about the creation and behavior of Glommed icons, see the section, “Creating a Glommed Icon” (p.387).

As an alternative to precise dragging, you could re-direct the creation of gloms by changing which subject was topmost, and which applicable stream was topmost — by vertically re-arranging the annotation within the stream or the streams within the Media Time Line. For more information about this, see the sections, “Moving Annotations Around” (p.392) and “Moving Streams Around” (p.383).

Cropping an Annotation

You can crop the end of an annotation by locating the Select-Bar at the point along the annotation at which you want to crop it, and then pulling the annotation’s Select-Bar icon off the Media Time Line onto the Desktop. Alternatively, you can select by clicking and then Cut (**⌘X**) the annotation’s Select-Bar icon, cropping the annotation’s end to the time-location of the Select-Bar. This last method can be particularly useful if there is a dramatic cut in the Movie; selecting All and then Cutting (**⌘A, ⌘X**) will crop **all** of the annotations currently intersected by the Select-Bar, cleaning the Media Time Line for the new and different annotations to follow.

Adjusting the End-Points of an Annotation

You can adjust the start-frame or the end-frame of an annotation by moving your cursor over the annotation’s endpoint: your cursor will change from the standard Arrow cursor to a Horizontal-Adjust  Cursor, indicating that, by depressing the mouse button, you will now be able to adjust the endpoint’s horizontal position. As long as you have a hold of the annotation’s endpoint, the Movie window will display the frame indexed by the time-location of that endpoint — giving you a convenient way of seeing the precise endpoint or startpoint of the feature of the Movie you are annotating. You can even drag the endpoint of the annotation off-screen, using the video shown in the Movie window to judge when the annotation’s validity commences or ceases. When you release the mouse button, the endpoint of the annotation will remain where you left it, even if that was in an off-screen position.

A useful shortcut for changing the length of an annotation is the **⌘E** key combination, which will pop the endpoint of a selected annotation to the position of the Select Bar. This can be even more expedient if, instead of scrubbing the Select Bar to the location you want, you pop the Select Bar's position to a desired horizontal location by clicking on a Media Time Line-stream's bounding frame.

Removing an Annotation

You can remove an annotation from the Media Time Line by clicking upon it to select it, and then Cutting it with the **⌘X** key combination. Alternatively, you can remove an annotation by dragging its Compound icon off the Media Time Line onto the Desktop.

Moving Annotations Around

Annotations in an expanded stream may be dynamically, vertically re-arranged, subject to an important restriction: An annotation cannot move in such a way that it would physically overlap or intersect another annotation. You can vertically re-arrange the annotations in an expanded stream by clicking upon and dragging the annotations to the height in their stream where you would like them to appear; the other streams will then shift their positions to accommodate your change if they are able. Some annotation movements may be constrained because neighboring annotations are themselves constrained from moving by other annotations.

Undo

Media Streams supports an unlimited multiple Undo, giving you the ability to undo the insertion, removal, and adjustment of annotations in the system, starting from the most recent. To undo your most recent operation, select Undo through the Edit Menu or with the **⌘Z** key combination. To undo operations prior to your most recent, select "Undo More" from the Edit Menu.

Operations with the Icon Palette

Making a Simple Query

A **Query** defines which icons are displayed in the Icon Palette. Thus, a query for "Objects" will return all of the Compound icons that contain an object. To make this simple kind of query, drag an icon into the Query Bar (from the Media Time Line, from a hierarchy in the Icon Workshop, or from the Objects Workshop

button at the top of the Icon Space); the icons returned will be all those Compound icons which either contain the elements of the query-icon, or hierarchical children of the elements of the query icon.

The Query Language

The Media Streams Query Language is a simple grammar for combining icons and filter units in the Query Bar in order to enable the construction of complex queries. You can re-arrange filter units in the Query Bar by grabbing their edges and moving them around.

- Two filter units side-by-side and unlinked to each other, or two icons located in the same filter unit (but unlinked to each other), are effectively treated as two separate queries. Media Streams reads them, however, as a single query for icons matching one set of conditions, “or” for another set of conditions. The number of “or”-ed queries you can make simultaneously is limited only by the width of the Query Bar. Each icon you drag into the Query Bar will receive its own filter unit, unless you drag it into the filter unit of another icon.
- Two filter units linked by an “and”  symbol, or two icons located within the same filter unit (but linked by an “and”  symbol) are treated as a single query for all those Compound icons which match **both** sets of conditions. To make an “and” query, drag one icon or filter unit over the other icon or filter unit until your cursor changes to the  cursor; releasing the mouse button in this state will create an “and” link between the two icons or filter units. To break apart the elements of an “and”-ed query, double-click on the  link. You can also make “and”-ed queries on the fly — as you drag new icons into the Query Bar — by dragging them over pre-existing query icons or filter units.
- Two icons or filter units, say X and Y, linked by the “temporal-overlap”  symbol will form a single query for all those Compound icons of type X which occur on Media Time Lines at the same time as an annotation of type Y. The icons returned by “temporally-overlapping” queries are always the same type as the **first** icon or filter unit in the “temporally-overlapping” expression. To make a  expression, drag the **first** icon or filter unit over the **second** icon or filter unit while holding down the  key; when you see the  cursor, release the mouse button and a new  query will be formed. Thus, to find the icons of “women” whose annotations are temporally-overlapping with Media Time Lines indexed by the “eighteenth century,” you would drop the “women” icon over the “eighteenth-century” while holding the  key; to do the reverse would return all those years in

the “eighteenth” century in which “women” occur. To break the  linker and split apart the query's components, double-click on it.

You can make compound expressions with the “and” and “temporal-overlap” linkers, using treating the filter units as a parenthesizing convention. In complex queries, the linkers follow an “Order of Operations”: The “and”  link is always performed before the “temporal-overlap”  link between filter units; the first element of a  expression determines the type of the result; and otherwise, queries are read left-to-right.

You can query for Movies, scenes from Movies, or annotationally-defined segments from Movies just as you would search for Compound icons. These queries can take the form of searches for all of the Movies, Scenes or parts of a Movie that are annotated by your icon of choice. Moreover, you can search for all of the icons used to annotate a specific Movie, or even all of the icons of a certain type used to annotate a specific Movie. These queries are performed by using Media Time Line Icons in combination with Compound icons on the Query Bar. Queries involving Movies are illustrated in the next section.

Some Representative Queries

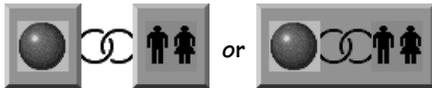
Reproduced below are examples of elementary query-types. The specific queries shown may or may not be useful to you; the intent of these examples, instead, is to demonstrate how filter units can be combined, and the meanings and results of their various combinations.



Query for all Compound icons containing elements of type X. This instance queries for all Compound icons that contain an object.



Query for all Compound icons containing elements of type X, **or** elements of type Y. These instances query for all Compound icons that contain either objects **or** characters. Were the filter units reversed, the results of the query would be presented in the opposite order.



Query for all Compound icons which contain elements of types X **and** Y. These instances query for all Compound icons that contain both an object **and** a character. The order of filter units in an “and” expression is unimportant.



Query for all Compound icons which contain elements of types X **and** Y **and** Z. This instance queries for all Compound icons that contain an object, a character **and** a character action.



Query for all Compound icons which contain elements of types X **and** Z, **or** Y **and** Z. This instance queries for all Compound icons that either contain a “land vehicle” and a screen position, or a “land animal” and a screen position.



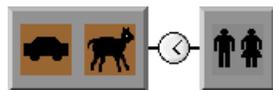
Query for all Compound icons which contain elements of type X, **or** which contain elements of types Y **and** Z. This instance queries for all Compound icons that either contain a “land vehicle,” **or** a “land animal” bound to a screen position.



Query for all Compound icons containing type X which appear on a Media Time Line at the same time as (i.e., which "**temporally-overlap**") annotations described by icons containing type Y. The queries shown here would return the icons of all “land animals” that appear on Media Time Lines at the same time as a character. The order of filter units in a "temporally-overlapping" expression is important: to query for all character icons temporally-overlapping with land animals, the filter units would have to be reversed. The first element of a “Temporally-overlapping” query determines the type of Compound icons returned.



Query for all Compound icons containing type X which **temporally overlap** annotations of type Y **or** Z. This instance queries for character Compound icons which appear on Media Time Lines **at the same time** as a "land vehicle" or "land animal" annotation.



Query for all Compound icons containing types X **or** Y which **temporally overlap** annotations of type Z. This instance queries for "land vehicle" **or** "land animal" Compound icons which appear on Media Time Lines **at the same time** as a character annotation.



Query for all Compound icons containing elements of type X which **temporally overlap** annotations of type Y, that **temporally overlap** annotations of type Z. This instance queries for all the object Compound icons which are temporally-overlapping with a character and a character action.



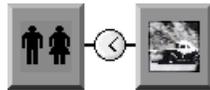
Query for all Compound icons containing elements of type X which **temporally overlap** annotations whose icons contain elements of types Y **and** Z. This query would return all "object" Compound icons that were temporally overlapped annotations defined by a character **and** a character action. In complex queries like this one, the "and" operator is performed before the "temporally-overlapping" operator.



Query for all Compound icons containing elements of types X **and** Y, that **temporally overlap** annotations described by icons containing elements of type Z. The query shown would return Compound icons containing an object **and** a character, that appeared on Media Time Lines **at the same time** as a character action.



Query for all the Compound icons **temporally overlapped** with the annotations of Movie A. The icon in the left-hand filter unit is a special character which stands for the set of all Compound icons. This query would return all the Compound icons used to describe the Movie represented by the Movie-thumbnail in the right-hand filter unit.



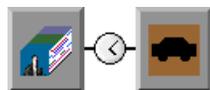
Query for all the Compound icons containing elements of type X **temporally overlapped** with the annotations of Movie A. This query would return all the character Compound icons used to describe the Movie represented by the Movie-thumbnail in the right-hand filter unit.



Query for all Compound icons containing elements of type X which **temporally-overlap** the annotations of Movie A **or** Movie B. The query shown would return all of the character Compound icons from either of the two Movies represented by the Movie-thumbnails in the right-hand filter unit.



Query for all Compound icons of type X which **temporally-overlap** the annotations of Movie A, **and** are also **temporally overlapped** with the annotations of Movie B. This query would return all of the character Compound icons common to the descriptions of Movie A and Movie B.



Query for all of the Movies described by Compound icons containing elements of type X. The icon in the left-hand filter unit is a special icon which stands for the set of all annotated Movies. This query would return, in the form of Media Time Line Icons, all of the Movies which at one time or another are annotated by a "land vehicle."



Query for all of the *segments* of Movie A which are annotated by Compound icons containing elements of type X. When a particular Movie is filtered in this manner, the results are Media Time Line Icons which represent the *segments* of the Movie which satisfy the query. The query shown would return, in the form of Media Time Line Icons, all of the *segments* of Movie A which are annotated by a "land vehicle."



Query for all of the **scenes** of Movie A which contain annotations labeled by Compound icons containing elements of type X. When a particular Movie is filtered in this manner, the results are Media Time Line Icons which represent the *scenes* of the Movie which satisfy the query. The query shown would return, in the form of Media Time Line Icons, all of the *scenes* of Movie A which contain annotations of a "land vehicle."



Query for all of the *segments* of Movies A which are annotated by Compound icons containing elements of type X, **or** all of the *segments* of Movies B which are annotated by Compound icons containing elements of type X. This query would return all of the *segments* from Movie A **and** Movie B which were annotated by a "land vehicle."

Sorting Your Results

You can sort the results of your queries in the following four ways:



alphabetically,



by type (i.e., position within the Icon Workshop hierarchy, and by the distinction between Glommed versus Compound icons),



by annotator,



by frequency of use.

If none of these are specified, Media Streams will sort the icons in the Icon Palette result-region by their recency of use.

Creating a new Icon Palette

It is possible to have more than one Icon Palette open at a time, each making different queries into the space of Compound icons. To create a fresh Icon Palette, go to the Media Streams Menu and select "New Icon Palette."

Alternatively, you can create a fresh Icon Palette by grabbing the rectangular edge of the Query Bar of any active Icon Palette. Your cursor will

change to a  hand as you pass over the Query Bar's edge, and when you click on the edge, your cursor will change to a  hand. As you pull your cursor away from the Query Bar, a dotted rectangular "ghost" of the Query Bar will follow; when you release your mouse button, this ghost will form the Query Bar of a new Icon Palette. You may find it advantageous to have several Palettes in use simultaneously, if you find yourself frequently re-using a number of icons retrieved from different queries.

A Media Streams Glossary

Annotation: a graphical descriptor of the Movie's content, comprised of a Compound icon and a color bar, and displayed inside annotation streams. An annotation is said to be valid over all the Movie frames indexed by its duration.

Compound icons: icons used to describe the content of the Movie. Compound icons appear on the Media Time Line (where they index states and events in the Movie), and in the Icon Space's Palettes (where they are stored and can be searched for). There are three types of Compound icons: Ordinary Compound icons, Glommed icons, and Compound Transition icons.

Animated Icon: a Workshop icon which uses a sequence of rapidly-animated frames in order to convey an idea. Most animated icons express character or object actions.

Animated Icon Editor: a tool with which you can edit pre-existing animated icons and create new ones. You can open the animated icon editor by option-clicking on an animated icon.

Color Bar: a horizontal colored stripe which extends from an annotation's Compound icon to the end of the annotation. Its color corresponds to the colors of the annotation's Compound icon. The length spanned by an annotation's Compound icon and color bar corresponds to the duration of the annotation's validity.

Compound icon: an icon which sits at the head of an annotation on the Media Time Line, labeling that annotation.

Compound Transition Icon: an Compound icon which describes cinematographic transitions from one shot to another. These icons contain a micon of the first shot, a Transition icon, and a micon of the second shot. Compound Transition icons are either generated in the Preprocessing or automatically constructed on the Media Time Line when a Transition icon is dragged from the Workshop to the Media Time Line.

Compound Filter Unit: two or more filter units which have been bound together by the  or  (“temporal-overlap” or “and”) filter-unit linkers. A compound filter unit acts as a single query.

Current Frame: the Movie frame currently displayed in the Movie window, whose time-index is shown in the time-index display, and whose content can be read in the select-bar icons.

End Frame: the last frame of the Movie for which an annotation is valid. It is marked by the right edge of the annotation's color bar.

Expandable Stream: a stream which can have multiple simultaneous annotations. Expandable streams will expand automatically when a new annotation is inserted into them.

Filter Unit: a small rectangular beveled-region in the Query Bar, which contains one or more icons, and is used to “filter” or query into the space of all Compound icons.

Glommed Icon: an Compound icon comprised of up to three Compound icons, each from potentially different Workshop hierarchies. Glommed icons are typically used to express a character or object's actions, screen positions, or relative positions; they are also used, however, to express Cinematographic framing, focus, camera-look-through, and tracking. Glommed icons are constructed on the Media Time Line and additionally appear in the Icon Palettes.

Gray Shadow Square: a gray shadow behind a Workshop icon used to indicate that that Workshop icon is the parent of a hierarchy with subordinate icons hidden underneath it. Double-clicking on a shadowed Workshop icon will open up its sub-hierarchy and hide its gray shadow square.

Hidebar: a bar at the bottom of the Media Time Line which contains hidden streams. There are two hidebars: one for audio and one for video. Hidden streams are represented on the hidebars by their stream control icons, and can be opened out onto the Media Time Line by double-clicking on their hidebar icons. Active streams can be hidden on a hidebar by double-clicking on their stream-controls.

Icon: an image representing an aspect of a Movie's content or a Movie itself. Compound icons are used to describe Movies; Workshop icons are the vocabulary from which Compound icons are made; and Media Time Line icons are three-dimensional extruded views which represent Movies or parts of Movies.

Icon Information Editor: an interface for augmenting and editing the content of an Compound icon, called up by double-clicking on the Compound icon.

The icon information editor allows you to establish the icon's title, and, depending on the icon type, its color or date. It also allows you to attach a textual comment to an icon.

Icon Palette: a part of the Icon Space interface which allows you to query for and retrieve Compound icons and parts of annotated Movies. The Icon Palette is comprised of the Query Bar and the Result region.

Icon Space: the interface containing the Icon Workshop and an Icon Palette, which allows you to create Compound icons and do searches into the space of Compound icons and annotated Movies.

Icon Title Editor: a facility of the Icon Information Editor which allows you to establish an icon's title and specify how it should appear when abbreviated, if necessary.

Icon Workshop: the hierarchically-organized, dictionary-like interface in the Icon Space, which contains the vocabulary of iconic elements which comprise Compound icons. The Workshop is also a facility for the construction of Compound icons.

Media Time Line: an interface in which time-indexed content descriptors of a Movie are arranged in streams.

Media Time Line Icons: three-dimensional extruded micons that represent segments of Movies. They appear only in the Icon Space's Palettes, where they are shown as the results of queries for Movies, and also where they can be used to further filter a query.

Movie: a media file containing digital video and/or digital audio.

Movie Controls: a set of tape-deck style controls which allow you to navigate through the Movie.

Movie Thumbnails: a set of Movie-frames subsampled at regular intervals, displayed in the Movie streams, which allow you to navigate the movie by double-clicking on them.

Movie Scrubbers: a pair of red rectangles which allow you to scrub through the Movie at greater scales than the Select Bar affords.

Movie Streams: a collective name for the two thumbnail streams and the videogram stream.

Non-expandable Stream: a stream which can only have one annotation at a time. Inserting a new annotation in a non-expandable stream will crop the ends of that stream's previous annotation to the Select Bar (if there is one), before inserting the new annotation.

Ordinary Compound Icon: an Compound icon comprised of up to three Workshop icons, each from the same Workshop hierarchy. These are created in the Workshop and can be found on the Media Time Line and in Icon Palettes.

Query: a search into the space of Compound icons for those icons which satisfy the conditions stipulated by the search. Queries appear as filter units on the Query Bar.

Query Bar: the rectangular region at the top of an Icon Palette, on which queries into the space of Compound icons are placed.

Result Region: the large rectangular region occupying the bottom of an Icon Palette, in which the Compound icons satisfying the conditions of the query in the Query Bar are displayed.

Result Sorts: the ways in which you can re-organize and re-display the results of a query: alphabetically, by annotator, by type, and by frequency of use. The result-sorts are controlled with the result-sorting buttons on the right edge of the Query Bar.

Shot Boundaries: shot boundaries are the points in the Movie determined by the Media Streams pre-processor to be “cuts” in the Movie. The default behavior for the “content jump” Movie Control, if no stream has been selected, is to advance through a Movie by its shot boundaries. Shot boundaries appear as abrupt discontinuities in the Videogram.

Scrubber: see Movie Scrubbers.

Select Bar: the thin gray line which runs vertically down the Media Time Line. The Select Bar is an indicator of the Movie's current frame and the current frame's content; it is also a scrub tool for scrubbing through small amounts of Movie data, and an insertion point and editing rule for new annotations on the Media Time Line.

Select Bar Icons: the Compound icons which float over the Media Time Line, attached to the right edge of the Select Bar. These icons display the content of the current frame indexed by the Select Bar, explaining the annotations the Select Bar crosses.

Start Frame: the first frame of the Movie for which an annotation is valid. It is marked by the left edge of the annotation's Compound icon.

Streams: streams are the time-indexed horizontal slots on the Media Time Line into which annotations are inserted. Each stream is devoted to containing annotations restricted to a specific aspect of the Movie's content.

Stream Controls: the hierarchically-organized beveled rectangles occupying a column on the left side of the Media Time Line. The stream controls govern which streams are visible, whether they are selected, and how they are vertically arranged.

Stream-Control Hierarchy: the hierarchical organization of the stream controls, which approximately mirrors the hierarchy of the streams and Icon Workshop.

Superordinate: a “parent” node in a hierarchy is said to be superordinate to the subordinate nodes beneath it.

Temporally-overlapping: one Compound icon is said to temporally overlap another if the first icon has an annotation which appears on a Media Time Line at some point with an annotation of the other. Put differently, two icons are temporally overlapping if there exists some part of an annotated Movie for which both icons' annotations are valid.

Terminal: a hierarchical node that has no “children.”

Time-Index Displays: a set of numeric displays located in the stream controls of the Movie Streams which report the minutes, seconds and frame-number of the current frame's occurrence in the Movie.

Time Line: see Media Time Line.

Videogram: a Movie stream which displays a thin slice taken from the center of every Movie frame, concatenated side by side. The videogram displays Movie data at the same scale as the Select Bar (one frame per four screen pixels) and enables a quick reading of how the Movie changes over a short range of time.

Workshop buttons: the row of fifteen buttons at the top of the Icon Space. The first thirteen of these buttons correspond to icon hierarchies, and are used to access those hierarchies in the Workshop. The rightmost two buttons stand for “the set of all icons” and “the set of all Movies,” and are used in special queries in the Query Bar.

Typical Work Flow.

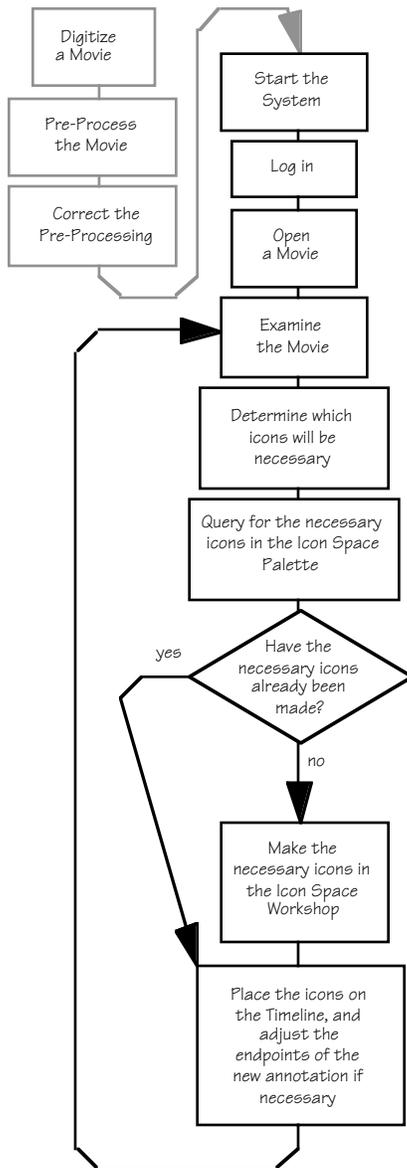


Figure 115.

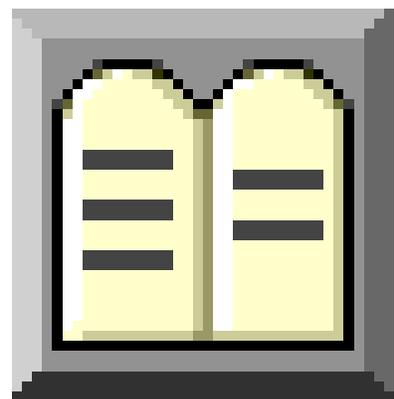
Annotation:

- On the Media Time Line, locate the Select Bar where you wish an annotation to start
- Query for appropriate descriptors in the Icon Space Palette, creating descriptors in the Icon Space Workshop if necessary
- Drag a descriptor from the Icon Space to the Media Timeline
- Trim or adjust the endpoints of the annotation if necessary

Dragging Icons in Media Streams

FROM:	Timeline	Icon Workshop	Select Bar	Query Bar	Select Bar Icon or Timeline Icon	Desktop
TO: Timeline Icon	Nothing.	The Workshop will open out to the components of the Timeline icon.	A duplicate of the Timeline icon will be created at the Select Bar in the same stream as the original, and a new annotation will be created.	A filter unit will be created in the Query Bar with the Timeline icon in it.	The dragged icon will glom onto the recipient icon if possible.	The Timeline icon and its annotation will be removed from the Timeline.
Workshop Icon	A single-element Compound Icon will be created from the Workshop icon and inserted at the Select Bar in the appropriate stream, and a new annotation will be created.	Nothing.	A single-element Compound Icon will be created from the Workshop icon and inserted at the Select Bar in the appropriate stream, and a new annotation will be created.	A filter unit will be created in the Query Bar with the Workshop icon in it.	The Workshop icon will glom onto the recipient icon if possible.	Nothing.
Select Bar Icon	If the Select Bar icon is dragged out of its stream, its annotation will be terminated at the Select Bar, removing the Select Bar icon from the Timeline for that frame. If the Select Bar icon is not dragged out of its stream, nothing will happen.	The Workshop will open out to the components of the Select Bar icon.	If the Select Bar icon is dragged out of its stream, its annotation will be terminated at the Select Bar, removing the Select Bar icon from the Timeline for that frame. If the Select Bar icon is not dragged out of its stream, nothing will happen.	A filter unit will be created in the Query Bar with the Select Bar icon in it.	The dragged icon will glom onto the recipient icon if possible.	The Select Bar icon's annotation will be terminated at the Select Bar, removing the Select Bar icon from the Timeline for that frame.
Icon inside Filter Unit inside Query Bar	The icon will be inserted at the Select Bar in the appropriate stream, and a new annotation will be created.	The Workshop will open out to the components of the icon from the filter unit.	The icon will be inserted at the Select Bar in the appropriate stream, and a new annotation will be created.	A duplicate filter unit will be created in the Query Bar	The icon from the Query Bar will glom onto the recipient icon if possible.	The icon and its filter unit will be removed from the Query Bar.
Icon in Palette Result-region	The icon will be inserted at the Select Bar in the appropriate stream, and a new annotation will be created.	The Workshop will open out to the components of the icon from the Result-region.	The icon will be inserted at the Select Bar in the appropriate stream, and a new annotation will be created.	A filter unit will be created in the Query Bar with the Palette icon in it.	The icon from the Result-region will glom onto the recipient icon if possible.	A pop-up dialogue will ask you whether you really wish to remove that icon from the set of all annotation icons
Icon Space Selector-Button Icon	Nothing.	The Workshop will open out to the hierarchy specified by the Selector-Button	Nothing.	A filter unit will be created in the Query Bar with the Selector-Button icon in it.	Nothing.	Nothing.

Part Three



Suggestions for Annotators

The Laws of Logging

The following brief guide outlines some recommendations which will help you ensure the repurposability of your annotations. It details various issues you should keep in mind when making and inserting annotations for selected types of media content.

The Five Commandments:

- Only that information which can be directly inferred should be annotated.
- If you can't see or hear it, don't log it.
- If it looks like a something, it is a something.
- If there's no icon for it, title its supercategory.
- Log the Audio and Video separately.



Inferable Time



Historical Period

The “Historical Period” stream refers not to the decade or century when the film was actually shot, but to the time-period the film purports to represent. Thus, the historical period of “Ben Hur” is the “Ancient” period, and not the 1950’s. In some cases, the two are the same; a clip from the 50’s, such as “Leave it to Beaver,” might be set in the 50’s. “Historical period” might even refer to the future (as in Star Trek) or to times preceding the evolution of humankind. Directors have used a number of cues to establish the historical period of a shot, including:

- period costumes and makeup
- period architecture and set-design
- period vehicles and other technology
- spoken accents

The historical period icon hierarchy is designed from a 20th-century, Western perspective. Thus, a video of a pre-industrial tribe in New Guinea, for example, should be annotated as the “Neolithic period,” unless there are clues (such as automobiles or hi-rise apartment buildings) to the contrary. Such annotation greatly

facilitates repurposing. The actual date of the recording of the video is annotated elsewhere, in the stream called “Timestamp.”



Season

In many cases, it is impossible to determine the time of year; spring, summer, winter and fall may all be easily mistaken for one another. We can, however, infer a shot's season from:

- a direct spoken or printed reference in the shot (calendars, newspapers)
- the type of clothing people are wearing (bikinis, parkas)
- the seasonal activities that people pursue (swimming, raking leaves)
- seasonally-attended locations (beaches, ski areas)
- the height of the sun in the sky (lower in winter)
- the foliage on trees, or the presence of flowers
- the presence of snow or other seasonal weather

No single one of these should be taken as a guarantee of a shot's season; rather, taken together, they allow the logger to make a good guess. If the season is not clear, or if the shot is an un-windowed indoor shot (without any of the above cues), then an annotation of the season is best omitted.

Certain parts of the world, regardless of their season, and because of their appearance, fit our idea of how a certain season “looks.” The season of a shot taken in the Caribbean region, for example — while actually filmed in the middle of the Caribbean “winter” — is perhaps best annotated as “summer,” according to the presence of the above-mentioned cues. Likewise, a shot of summer in the Arctic, with children playing in snow, is best annotated as a “winter” shot. For this reason, it is logical to equate “season” with “climate” when annotating shots taken at such locations.



Time of the Day

It is relatively easy to determine, given an outdoor shot, whether it is daytime or nighttime. And, generally, that information is sufficient for most purposes. Some confusion may arise, however, in the annotation of the finer gradations of the time of the day. Sunrise and sunset, for example, may look virtually identical. Indoor shots without the affordance of windows likewise present an opportunity for confusion. In determining the time of day in cases like these, it is

wise to use the information provided by a variety of external cues, such as:

- a direct spoken or printed reference (clocks, watches)
- the activities pursued by people that generally correlate to specific times of the day (breakfast, lunch, dinner, work, awakesness, sleep)
- the crowing of roosters, or the sounds of rush hour
- clothing specific to certain times of the day (nightgown)



Inferable Space



Geographical Space

Geographical location refers to the place in the world where the shot appears to be set. Under many circumstances, the geographical location of a shot will be a city, state, or country. Sometimes, the location can be inferred indirectly from cues like:

- direct spoken or printed reference (captions, signs)
- the language spoken, and its accent or regional dialect
- the ethnicity of the people, their regional costume, architecture, and technology
- the climate, topography, flora and fauna
- the written alphabet used, when visible
- familiar landmarks
- regional music

Oftentimes, however, there is insufficient visual information to infer the precise geographical location. As usual, only that information which can be directly inferred should be annotated. The location of a shot of a man marooned on an island, therefore, might simply be “an island”; of a woman in an alley, “a city”; of a plantation, “South-eastern United States”; of a man on a raft in the ocean, “an ocean” — without specification as to which city, or which ocean, or in what part of the world, because such information cannot be determined. The actual location of the recording of the video is annotated elsewhere, in the stream called “Spacestamp.”

Some indoor shots, even those that have several of the above cues, could take place anywhere — on land, at sea, or even in

space. If this is the case, it is best to omit a description of geographical location.



Functional Space

Functional space divides the location of shots into those which occur in or around buildings, public outdoor spaces, wilderness, and vehicles. A scene might transpire in a car, submarine or building, on a highway, on a baseball field or in a jungle. When the location of a shot combines functional spaces, ambiguities can arise in which the annotator must distinguish between what is truly the functional space, and what may be better described as architectural or natural objects. Consider the following two scenes:

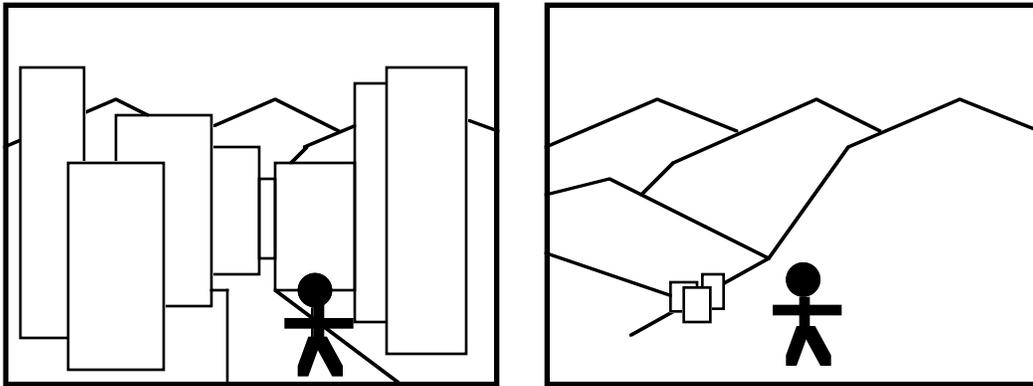


Figure 116. Functional Space can depend on predominance.

Although both scenes feature mountains and buildings, the functional setting of the first scene is a public outdoor space — the street — whereas the second scene's functional setting is the wilderness of the mountains. Put another way, a Media Streams annotator would describe their settings, respectively, as, “on top of **a street** in a city” and “on the side of **a mountain** on land.” Moreover, the mountains in the first scene, and the buildings in the second scene, would be described as natural and architectural objects. Why isn't “mountain wilderness” the functional setting of scene 1? To understand why, it is useful to examine a scene with the following questions in mind:

- Where is the major action taking place? What is the setting of the **action**?
- Which setting is visually and proportionally dominant?

- What aspects of the landscape do the actors interact with as props, versus those aspects that are scenery, and those that are setting?
- Do the natural or architectural objects in the vicinity have such power of recognizeability that they are better labeled as functional locations (for example, a generic building vs. McDonald's; mountain vs. Mt. Rushmore) than as objects?

In scene 1, the functional location of the action is a street, and the geographic location of that street is a city. The mountains in the background are “natural objects” because, as with clouds, one could imagine removing them from the scene (or even do so, with the proper technology) without detracting from our understanding of the scene's functional location. In scene 2, a person is functionally set in the mountains. Since the buildings in the background are generic examples of their object-class, we label them as architectural objects as well. If, however, the buildings were a well-known landmark, such as the Potola Monastery, we might instead describe scene 2 as “in front of a **religious facility** in Tibet.” Notice how many landmarks, such as the Eiffel tower or Mount Rushmore, carry with them an indication of or pointer to their geographical location.



Topological Space

Complete descriptions of a shot's location provide a geographical location, a functional location, and a topological relation to that functional space. Indoor shots typically use variations of the “inside” topological descriptor, e.g., “**by the rear left corner inside** a restaurant in the Midwest.” The topological descriptions of outdoor shots are usually expressed with respect to the buildings or large natural formations that characterize the functional location, e.g., “**in front of** a restaurant in the Midwest,” “**above** a street in New York,” or “**on top of** a mountain in China.”

Topological relations assume but do not require that their object has an aperture from the outside to the inside, and that the “front” of their object is the space in front of this aperture. For the special case of scenes which appear to be set outside the rear or alternative exits of a building, annotators should be careful to indicate that the scene is in fact set “behind” the building, though apertures might be present in the objects context. All scenes which occur on the threshold of a building's aperture, regardless of

whether it is a “rear” or “front” aperture, however, should be described as occurring “on the threshold.”

One potentially confusing aspect of topological space descriptions is that they describe **where the action is set** in relation to some functional description, and not **where the camera is set** in relation to some functional description. That particular aspect of video content is annotated separately in the “camera relative position” context. Thus, if the principle characters in a scene are dancing on a building’s roof, even though the scene is shot from some camera-position on the ground, the topological location of the scene is “**on top of a building.**”



Weather

Generally speaking, weather should only be annotated if the scene is shot outdoors, or if the outdoors is visible through a window or other aperture. Even if Chris were to walk in, covered with snow, while brandishing a snow shovel, it would not be legitimate to annotate this un-windowed indoor scene with weather. If rain is audible in the soundtrack, but not visible in the video, only the Audio Weather track should be annotated with rain.

While the amount of precipitation or clouds in a shot's weather are easily readable, the wind level can only be inferred. Typical windiness cues are:

- the howl of wind (for the audio:weather context only)
- hair, clothing or foliage blown by the wind
- loose objects blown adrift



Timestamp (Actual Time)

The Timestamp stream is used to annotate the actual date and time on which a scene was filmed. By way of contrast with the Inferred Time stream, nearly all Timestamp annotations are likely to contain some specification of the Twentieth century, as nearly all recorded sound or moving images have been produced in the past hundred years. Timestamp annotations use the same hierarchy of descriptors as are used for Inferred Time — but the difference is

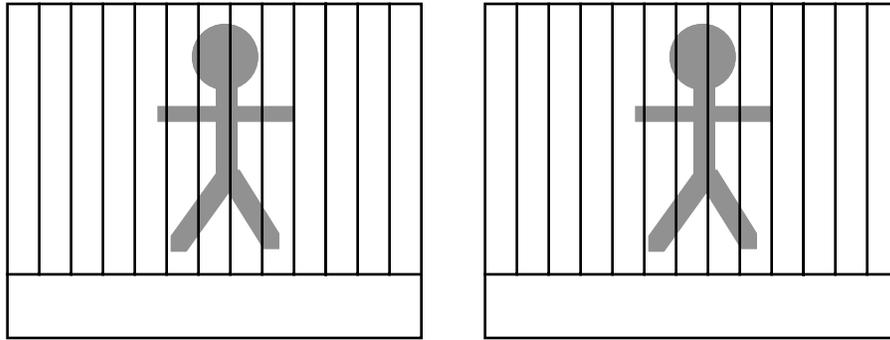
that Inferred Time annotations are what appear to be the case, while Timestamp annotations are what we **know** to be the case. Making this distinction facilitates flexible re-purposing while preserving the integrity of the media's true source. It is sufficient to annotate the Timestamp with incomplete information, such as a decade-annotation without the specification of a year, if that is all that is known.



Spacestamp (Actual Location)

The Spacestamp stream is used to annotate the actual location at which a scene was recorded. Spacestamp annotations use the same hierarchy of descriptors as are used for Inferred Space — but the difference is that Inferred Space annotations are what appear to be the case, while Spacestamp annotations are what we **know** to be the case.

Buster Keaton makes an interesting play on the distinction between inferred space and actual space in his short movie “Cops”. The movie opens with a close-up of a dejected Keaton talking to his sweetheart from behind bars. By all appearances, Keaton is incarcerated. Gradually, however, the trucking of the camera reveals that Keaton is actually outside the gate of his sweetheart's estate. In this case, the Spacestamp of the scene would be “behind a gate,” while the annotation of the inferred Space would **change** from “inside a jail” to “behind a gate” when a jail was no longer a visually plausible description of the scene. The advantage of annotating both the Inferred Space and the actual space (the Spacestamp) is that scenes like Keaton's could then be repurposed in **multiple ways**.



A man in jail

A man behind a gate

Figure 117. Actual versus Inferable locations



Characters

The most important rule about the annotation of characters is that, when a character leaves the screen, her annotation must leave with her. The moment a character is no longer visible in the Movie frame, her annotation is invalid. If a character does not appear on the screen, but is still audible in the soundtrack (as frequently happens with narrators and interviewers), then the character can be legitimately annotated in the Audio Characters stream — but not in the Video Characters stream.



Gender and Age

The first branch of the characters hierarchy specifies a character's gender and age, and/or whether the character is of a non-human variety. "Human" is currently meant to be understood in a very strict sense, insofar as cartoon characters and puppets, even in human form, are excluded from this designation. If you wish to title a character's icon, try (if at all possible) to use the character's actor's name.

Ordinarily inanimate objects and animals which act as characters (as can happen in cartoons, e.g., **The Happy Toaster**, **The Lion King**) are not well-supported by the Media Streams character hierarchy. Although it would not be incorrect to label a toaster-character as a "non-human character," it is strongly recommended that this designation be reserved for the organic and phantasmal beings (such as aliens and ghosts) which fill out its hierarchy. Instead, animal-characters and object-characters (the

Happy Toaster, Talking Diaper, etc.) are best labeled as named instantiations of their object class; moreover, like other objects, they may be annotated with the ordinarily human actions that we associate with characters (talking, walking, etc.).



Occupation

The second branch of the character hierarchy is “occupation,” also called “role.” Annotating a character's role can be highly confusing because it must be inferred from evidence which, in reality, is completely independent from a character's identity. The most important cues for determining a character's role are:

- what the character is wearing (uniforms, costumes)
- the tools or other objects that are manipulated by or co-present with the character
- what the character is doing
- the character's immediate setting

Taken independently, none of these are sufficient indicators of a character's role. Merely proffering a hamburger or tossing a baseball, for instance, does ensure that a character is a “fast-food worker” or a “baseball player;” it could simply be that a character has proffered a hamburger or tossed a baseball. A person sitting in a jail cell or behind a desk is not necessarily a criminal or a receptionist. Likewise, the holding of a wrench is not a guarantee that the holder is a repair-person. In general, the best indicator of a character's role is the costume they wear. But plenty of roles have no established, identifying “costume.” And in other cases, the fact that a character is wearing a particular costume is better annotated as the wearing of that costume as clothing in the Characters Actions stream, rather than interpreted as an indication of that character's role.

To solve this conundrum, we must rely on our stereotypes of what people in certain occupations or with certain roles are like, and solve an equation which has variables in several different streams. It is generally easy to discern police from other characters, not just by their uniforms and paraphernalia, but also by their characteristic actions (directing traffic, displaying badges, firing handguns) and often by their settings (police stations, police cars, streets). Firemen, chefs, surgeons, soldiers, hippies and priests are likewise readily identifiable. But if, on the other hand, a policeman were to go home, take off his uniform, remove his gun and

sit before the television in his underwear — eliminating all cues as to his occupation — it would be incorrect to continue annotating him as a policeman. Instead, he would be a man with no discernible occupation, lounging in his underwear. A character's role may change or be eliminated altogether in as little time as it takes to remove a hat.



Number/Configuration

The last branch of the character hierarchy labels the number of characters *seen together* who share the same gender/role makeup. Characters should only be grouped into numeric compounds if they share the same makeup **and act in unison**. So grouping them has the advantages of saving valuable screen space, as well as offering enhanced retrieval and more accurate descriptions of group or crowd actions. Often, a single member of a group will act apart from his or her role-mates, as when an individual steps forth to act as a spokesperson. To facilitate the proper construction of action-gloms, the **n** characters present in such an example are best annotated as a group of **n-1** people and a solitary individual with the same role as the others. It is legitimate to annotate a group of people with just a number, if their genders and roles are indeterminate. If their number, too, is unknown, it is sufficient to annotate them with the label “many,” or with one of the crowd-icons that hint at the crowd size and density.



Objects

The most important rule about the annotation of objects is that, if an object moves out of view of the camera, then its annotation must leave with it. The moment an object is no longer visible in the Movie frame, its annotation is invalid. If an object does not appear on the screen, but is instead audible in the soundtrack (e.g., a phone ringing), then the object can be legitimately annotated in the Audio Objects stream — but not in the Video Objects stream.



Body-part Objects

Although the annotator should find the logging of objects to be straightforward, some confusion may arise when the object is a part of the human body. A special category of objects called body-part objects has been created to handle these occasions. Icons in this category will find their clearest utility when there is a disembodied body-part on the screen, such as a brain in a jar. But suppose instead we have a close-up of a hand which is obviously, owing to its activities, controlled by some living person. When the body-part is attached to a living character we are forced to decide whether we are looking at a close-up on a character or a close-up on a body-part object. The answer is that, if the body-part reveals no clue as to the identity of its owner, then it is an object (capable, naturally, of human actions). According to this scheme, therefore, it is unlikely that a close-up on a character's face would be considered a shot of a body-part object, except under extreme magnification. If the body part's owner is identifiable, but the part is disembodied, the annotator should title the body part with its owner's name in the possessive case (e.g., “**Medusa's** head”).



Furniture, Indoor Architectural Objects and Appliances

A distinction is made between **furniture-objects**, which (at least in principle) could be moved or replaced, and **indoor architectural objects**, which are thought of as permanent fixtures of an environment. Thus, examples of furniture-objects are chairs, beds, and tables, while examples of indoor architectural objects are light-switches, electric outlets, windows, doors, and faucets. **Appliances** are a sub-category of **tool-objects** whose domain occasionally overlaps some indoor architectural objects (such as stoves or sinks), but also includes other stationary indoor artifacts (like toasters) which cannot truly be thought of as architectural.



Relative Positions

The yellow relative position icons are used in four different contexts: in the topological hierarchy of the **space** context, in which they qualify the scene's relationship to its functional and geographical setting; in the **character relative-positions** context; in the **objects relative positions** context; and in the **camera relative-positions** context. In the latter three contexts, the relative positions icon express the physical relationship of a character, object or the camera (represented by a pink sphere) to another character or object.

This latter character or object is represented in the relative-positions icons by a boxlike container, in order to accommodate possible relationships like “inside” or “on the threshold of.” External relative-positions should be annotated with respect to the latter object's customary orientation, if it has one.

It is possible for a character, object or the camera to participate in more than one relative-position relationship. For example, a man could be inside a truck, while also on top of a chair. It is especially important to annotate relative positions that are explanatory (such as “driver inside car”), or seem out of the ordinary (such as “woman on top of refrigerator”).



Character Actions

Many Character Actions (and Object Actions) are **transitive** — that is, they can have a grammatical “object” in addition to their subject and verb. Examples of action expressions with grammatical objects are “Pat is eating **pizza**,” “Chris punches **Toby**,” or “Jean is walking **to the right**.” Some expressions even require an object to make sense, such as “Bert looks at **Ernie**.” Media Streams uses an English-like grammar (SVO) for Character Action and Object Action expressions, concatenating the “object” of the action (which may in fact be an object, character, or direction) onto the right-hand side of the expression.

To specify that the object of an action is a direction, annotators should use the Screen Position icons to complete the action expression. This is because, in addition to their use in describing the screen positions of characters and objects, the

Screen Position icons can also be understood to describe direction vectors that begin at the center of the screen and point outwards through their blue indicators.

There are many occasions on which characters will perform or undergo actions which are best characterized as Object Actions — that is, the character's body will behave **like an object**. Humans can fall, bounce, or break apart in much the same way as objects, and, in cartoon form, are capable of nearly unlimited topological and material acrobatics. Such behavior is not even limited to cartoons: the T-1000 robot from **Terminator 2**, for example, performs a characteristic Object Action when he melts into a puddle of goo. Media Streams supports the unrestricted annotation of characters with Object Actions, and vice versa (for occasions when objects, like the Happy Toaster, exhibit all-too-human behaviors). Generally speaking, Character Actions are volitional acts while Object Actions are not — but there are plenty of exceptions to this. The distinction between Character Actions and Object Actions is purely pragmatic.

Many branches in the Characters Action hierarchy are divided into **conventional** and **abstract** motions. Conventional motions are the most common and meaning-laden motions of our lives, such as walking, smiling, or clapping; abstract motions, on the other hand, are the pure rotations and motions of our joints and limbs, to which (ordinarily) little or no meaning is attached. An example of an abstract motion might be the flexing of the elbow or the swiveling of the arm about the shoulder.



Object Actions

As with Character Actions, many Object Actions are transitive and can take a grammatical “object” (another object, a character, or a direction) in annotations. Object Actions, like Character Actions, can be used to describe the behavior of both Objects and Characters. Object actions are divided between **Motions** and **State Changes**.



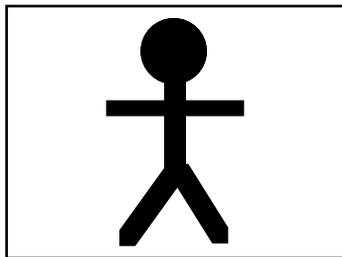
Cinematography



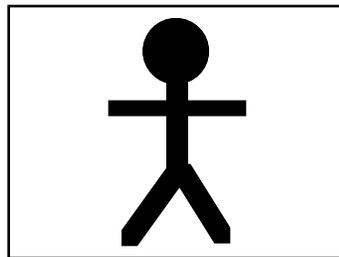
Framing

If the primary action in the scene concerns an object, annotators should take care to describe the framing using “object framing” icons. If the primary action involves people, or a mixture of people and objects, annotators should use the “human framing” descriptors instead. If the primary action involves multiple characters or objects, each with different framings, the default is to describe the framing of the one closest to the camera.

These framing descriptors should not be interpreted as additionally describing an object's shape, actual size or screen position; they merely describe the relationship of the object to the rectangle of the movie window, expressed as a ratio of the object's distance from the camera to its actual size. Thus, a small children's doll and an adult human, if they both occupy the same proportion of the screen, would be annotated with equivalent framings — even though the camera is focused at different distances:



a medium shot
of a child's doll



a medium shot
of Andre the Giant

Figure 11B.

The framing of a shot can change over time, owing to changes in the position of the camera (truck motions), changes in the setting of the camera's lens (“zoom”), or simply because the characters or other subject matter in the scene have moved closer to the camera.



Appendix B

**Code Listing for Media Streams
Retrieval Algorithm**

Portions of Retrieval Code



```
;;; get-similar-sequences  
;;; Top level function for finding sequences similar to a query time line  
;;; - enumerate temporal relations so we can easily calculate the temporal  
;;; constraints on the query  
;;; - use Mnemosyne indices to find matches for the compounds in the query  
;;; - loop through all shots in temporal order  
;;; - determine which compounds are visible within the current shot  
;;; - calculate the temporal constraints between those compounds  
;;; - create a vector, whose length is equal to the number of compounds within the shot  
;;; each element of the vector is a vector of possible matches for the compound, sorted in  
;;; order of semantic similarity, and "clipped" to a certain semantic threshold.  
;;; - the query is used as a template, which we then try to fill with the best matches which  
;;; are temporally consistent, and accumulate these best templates into a list  
;;; - we then combine the template matches for each shot to create the highest scoring combined  
;;; sequences, removing sequences with a zero score. Only calculate the top matches-threshold  
;;; combinations.
```

```
(defun get-similar-sequences (query-time-line)  
  (let ((query-time-line-frame (framer-frame query-time-line))  
        (matches-threshold (matches-threshold query-time-line))  
        (semantic-score-threshold (semantic-threshold query-time-line)))  
    (enumerate-temporal-relations-in-time-line-frame query-time-line-frame)  
    (let ((hit-hash-table (find-similar-compound-icon-frames query-time-line))  
          (shots (sort (copy-list (framer::frame-annotations (get-scene-bucket query-time-line-frame)))  
                       #'< :key #'start-frame))  
          (shot-number 1)  
          (shot-matches nil))  
      (format t "~%Finished Mnemosyne search.")  
      (dolist (shot shots)  
        (format t "~%Matching shot ~D" shot-number)  
        (incf shot-number)  
        (let ((visible-compounds-in-shot nil)  
              (completeness-threshold 0))  
          (dolist (component-frame (frame-ground shot))  
            (let ((compound (find-compound-from-component-frame component-frame)))  
              (when (view-window compound)  
                (pushnew compound visible-compounds-in-shot))))  
          (setf completeness-threshold (floor (* (completeness-threshold query-time-line) (length visible-compounds-in-shot)10))  
                (let* ((blank-template (make-template visible-compounds-in-shot))  
                      (results (find-best-templates visible-compounds-in-shot  
                                                    (convert-hit-hash-table-to-vector-of-vectors visible-compounds-in-shot  
                                                    hit-hash-table semantic-score-threshold  
                                                    (temporal-constraints blank-template)  
                                                    matches-threshold  
                                                    completeness-threshold)))
```

```

(push (if results
      results
      (list (make-template nil)))
      shot-matches)))
(setf shot-matches (nreverse shot-matches))
(format t "~%Combining shots")
(let ((matches-thresholded-scored-combinations (combine-templates-for-shots shot-matches matches-
  threshold)))
  (delete-if-not #'(lambda (movie-sequence)
                    (score> (score movie-sequence) *null-score*))
                (mapcar #'(lambda (scored-combination)
                            (let* ((templates scored-combination)
                                   (intermediate-movie-sequences
                                    (mapcar #'make-movie-sequence templates))
                                   (merged-movie-sequence(merge-movie-sequences
                                                           intermediate-movie-sequences)))
                              (setf (templates merged-movie-sequence) templates)
                                  merged-movie-sequence))
                          matches-thresholded-scored-combinations))))))

```

;;; **find-similar-compound-icon-frames**

```

;;;      uses Mnemosyne indices to find compounds which have components which are similar to
;;;      components in the query
;;;      accumulates the result in a hash table which is indexed by the compounds of the query,
;;;      whose values are lists of matching compounds
;;;      - start-search primes the Mnemosyne matcher with the top level of the structure for
;;;      which we are searching
;;;      - continue-search performs the actual search, which we only do if the component is
;;;      within the supplied temporal extent and in a visible context
;;; we then reindex the search hash table so that its values are sorted and ready for temporal comparisons

```

```

(defun find-similar-compound-icon-frames (media-time-line
                                          &optional start-frame end-frame
                                          &key (only-visible-time-line-contexts t))
  (let* ((media-time-line-frame (framer-frame media-time-line))
         (temporal-extent-supplied-p (and start-frame end-frame)))
    (start-search media-time-line-frame)
    (do-features (component media-time-line-frame)
                 (unless (and temporal-extent-supplied-p
                              (not (temporally-conjunctive? component start-frame end-frame)))
                     (unless (and only-visible-time-line-contexts
                                   (not (in-visible-context-p component media-time-line-frame)))
                         (continue-search component))))))
    (reindex-search-hash-table *current-search-table*))

```

```

;;; start-search
;;; setup some globals for Mnemosyne

(define-dual-function start-search (frame)
  (setq *current-search-head* frame)
  (setq *current-search-table* (make-hash-table)))

;;; continue-search
;;; calculate how deep the current frame is below the search head
;;; and mark similar frames using this depth information

(define-dual-function continue-search (frame &optional (priority 1))
  (do ((f frame (frame-home f))
      (d 0 (1+ d)))
      ((or (null f) (eq f *current-search-head*))
       (if (null f) (error "~S isn't beneath the current search" frame)
           (mark-similar frame priority d *current-search-table*))))))

;;; mark-similar
;;; a modified Mnemosyne function
;;; if the query frame has a ground, we check its index
;;; for similar relations and mark them in the search table
;;; we then mark-similar for all of the annotations on the
;;; query frame
;;; mark-home
;;; this is the actual function which is called when
;;; for-coindices finds a similar relation
;;; in general, it scores the match,
;;; unless the types of the compounds are different
;;; and not characters-action and objects-action
;;; (which should be able to match)
;;; or we have already scored this compound for this
;;; type of relation (*current-mark-table* is only valid
;;; during the for-coindices call)

(defun mark-similar (query-frame priority depth table)
  (flet ((mark-home (frame)
          (let* ((compound-of-query (ms::find-compound-from-component-frame
                                     (frame-home query-frame)))
                (compound-of-match (ms::find-compound-from-component-frame (frame-home frame)))
                (type-of-query (type-of compound-of-query))
                (type-of-match (type-of compound-of-match)))
            (unless (or (and (not (or (and (eq type-of-query 'characters-action-compound-icon)
                                           (eq type-of-match 'objects-action-compound-icon))
                                     (and (eq type-of-query 'objects-action-compound-icon)
                                           (eq type-of-match 'characters-action-compound-icon))))
                        (neq type-of-query type-of-match))
                    (gethash compound-of-match *current-mark-table*)))

```

```

(setf (gethash compound-of-match *current-mark-table*) T)
  (let* ((compound-hit-score-assoc-list (gethash compound-of-query table))
        (current-score-with-components (cdr (assoc compound-of-match compound-hit-
          score-assoc-list))))
    (if current-score-with-components
      (score-match query-frame frame current-score-with-components)
      (let ((blank-score (list (make-instance 'score))))
        (setf (gethash compound-of-query table) (cons (cons compound-of-match blank-
          score) compound-hit-score-assoc-list))
        (score-match query-frame frame blank-score))))))
(unless (empty-p (frame-ground query-frame))
  (my-for-coindices #'mark-home query-frame))
(do-features (f query-frame)
  (mark-similar f priority (1+ depth) table)))

```

;;; **my-for-coindices**

;;; set up a **current-mark-table**, which keeps track of
 ;;; matches made for this relation
 ;;; since we only score things once per relation, the mark
 ;;; table is moved outside of the do-prototypes loop
 ;;; (originally **current-mark-table** was bound within the loop)
 ;;; we loop over the prototypes of the relation and try to
 ;;; find matches in the index which have similar grounds
 ;;; to the query relation (calling *my-get-bucket*)
 ;;; if it turns out that we are checking a relation in a *cidi*
 ;;; above the *cidi*-prototype terminals, then call *my-for-coindices*
 ;;; recursively on the same relation for all the frames of the subordinate
 ;;; icons of the *cidi*, passing along the original-relation-frame, whose
 ;;; ground we are interested in

```

(defun my-for-coindices (function frame &optional (original-frame frame))
  (let ((*current-mark-table* (make-hash-table :test #'=))
        (check-index-from (if (fast-cidi-frame-p (frame-home frame))
                              frame
                              (frame-prototype frame))))
    (do-prototypes (p check-index-from)
      (do-results (g (frame-ground original-frame))
        (let ((buckets (my-get-bucket p g)))
          (when buckets
            (dolist (bucket buckets)
              (do-results (c (frame-ground bucket))
                (unless (frame-deleted-p c)
                  (funcall function c))))))))
    (when (above-terminal (frame-home check-index-from))
      (dolist (subordinate-icon (ms::subordinate-icons
        (ms::get-clos-object (frame-home check-index-from))))

```

```
(let ((same-relation-under-subordinate-icon
      (find-named-annotation (%frame-name check-index-from) (ms::framer-frame subordinate-icon))))
      (my-for-coindices function
                        same-relation-under-subordinate-icon
                        original-frame))))))
```

;;; my-get-bucket

;;; similar to original get-bucket, except this version
;;; recurses for stuff above prototype terminals
;;; given a relation and a ground, return the list of
;;; frames in the index which are grouped under the
;;; same prototype terminal as the given ground
;;; if the terminal is above the cidi-prototype-terminals
;;; then recursively call my-get-bucket to collect all
;;; the things indexed under the same relation for
;;; all of the subordinate-icons of the terminal

```
(defun my-get-bucket (frame ground &optional (create NIL))
  (let ((index (if create
                   (make-annotation frame "+index")
                   (local-probe-annotation frame '+index)))
        (terminal (if (frame-p ground)
                      (get-terminal-prototype ground)
                      ground))
        (result nil))
    (when (and (fast-cidi-frame-p terminal)
              (above-terminal terminal))
      (dolist (subordinate-icon (ms::subordinate-icons (ms::get-clos-object terminal)))
        (setf result (append result
                             (my-get-bucket frame
                                           (ms::framer-frame subordinate-icon))))))
      (when index
        (let* ((alist (rs->list (frame-ground index)))
              (pair (if (frame-p ground) (assoc terminal alist)
                       (assoc ground alist :test #'equal))))
          (if pair
              (setf result (push (cdr pair) result))
              (and create
                   (let ((bucket (make-unique-annotation index "v")))
                     (nadd-to-set index (cons terminal bucket)
                                   bucket))))
              result))
```

```

;;; score-match
;;; calculates the score of a match given the query relation and match relation
;;; and destructively modifies the score-with-component-hits total
;;; query-relation and match-relation are identical, but we need to compare the
;;; components between which the relation exists
;;; score-with-component-hits is a list whose first element is the current score
;;; and whose rest is a list of components which have matched against this query
;;; if the relation is a temporal relation, we increment the score by *temporal-match-score*
;;; since we are not currently indexing temporal relations, this case will not be invoked
;;; otherwise
;;; we check to see if we have already scored this match
;;; if we haven't, then compare the cidis of the homes of the relations
;;; normally, the cidi is simply the frame prototype of the component frame in
;;; a time line (which is how we get the cidi of the query), but since prototypes
;;; can be shifted in the logs, we don't assume that for the match component
;;; (we instead call get-cidi-of-match, which checks to see if the prototype has
;;; been shifted to a component on a time line)
;;; we then compare the cidis of the sets of grounds (a n x m comparison, although
;;; most of our relations have single valued grounds)

```

```

(defun score-match (query-relation match-relation score-with-component-hits)
  (let ((current-score (first score-with-component-hits)))
    (cond ((ms::temporal-relation-p query-relation)
           (add-scores current-score *temporal-match-score*))
          (t
           (let ((match-frame (frame-home match-relation)))
             (when (dolist (el (rest score-with-component-hits) t)
                           (when (eq match-frame el)
                               (return nil))))
               (push match-frame (rest score-with-component-hits))
               (let* ((query-frame (frame-home query-relation))
                     (cidi-of-query (frame-prototype query-frame))
                     (cidi-of-match (get-cidi-of-match match-frame cidi-of-query)))
                 (add-scores current-score (score-cidi-match cidi-of-query cidi-of-match))))
             (do-results (match-ground (frame-ground match-relation))
                         (when (dolist (el (rest score-with-component-hits) t)
                                         (when (eq match-ground el)
                                             (return nil))))
                           (do-results (query-ground (frame-ground query-relation))
                                       (let* ((query-ground-cidi (frame-prototype query-ground))
                                             (match-ground-cidi (get-cidi-of-match match-ground query-ground-cidi)))
                                         (add-scores current-score (score-cidi-match query-ground-cidi match-ground-cidi))))
                                       (push match-ground (rest score-with-component-hits))))))))))

```

```

;;; score-cidi-match
;;; compares two cidis and calculates a score
;;; if either cidi is a named character we do a special case
;;; if both are named and they are named the same, then
;;; return a better than perfect score (2 exact points)
;;; if one is named, still return an exact match if it is a
;;; a named version of the other
;;; this distinction is necessary because John the adult male
;;; should match an adult male perfectly, while in general,
;;; titling an icon is equivalent to making a further specification
;;; and is treated just as if it were one level down in the cidi
;;; hierarchy
;;; special casing characters was found to be useful empirically, and
;;; we may find that titling should be treated differently in general
;;;
;;; if we have named characters, for sake of further comparison,
;;; use the unnamed versions. Thus, John the male will be a
;;; prototype of adult male, and not a sibling.
;;;
;;; if neither are titled characters, we do the standard comparison
;;; if the two cidis are identical, return an exact match, and weight
;;; actions higher. In general, continuity of action is more important
;;; than continuity of actors
;;; if the query is a prototype or a home of the match, return q-is-proto with
;;; a value equal to the difference in depth
;;; if the match is a prototype or a home of the query, return m-is-proto with
;;; a value equal to the difference in depth
;;; if the two are siblings, return a siblings score of 1
;;; if the two share some prototype (which they usually will because
;;; of the indexing mechanism), return bad match of 1
;;; otherwise return a null score
;;; we need to check both for prototype and home depth, because queries
;;; above cidi-prototype-terminals are in effect prototypes of their sub-annotations
;;; which have been severed for efficiency reasons.

```

```

(defun score-cidi-match (query-cidi match-cidi)
  (let ((named-character-match (and (has-home match-cidi ms::*character*)
                                     (not (local-probe-annotation match-cidi '+icon-name))
                                     (frame-prototype match-cidi)))
        (named-character-query (and (has-home query-cidi ms::*character*)
                                     (not (local-probe-annotation query-cidi '+icon-name))
                                     (frame-prototype query-cidi))))
    (cond ((and named-character-match named-character-query
                (eq query-cidi match-cidi))
           (make-instance 'score
                          :exact 2))
          ((and named-character-match (eq named-character-match query-cidi))
           (make-instance 'score
                          :exact 1))
          (t nil)))

```

```

(and named-character-query (eq named-character-query match-cidi))
  (make-instance 'score
    :exact 1))
(t
  (when named-character-match
    (setf match-cidi named-character-match))
  (when named-character-query
    (setf query-cidi named-character-query))
  (if (eq query-cidi match-cidi)
    (make-instance 'score
      :exact (if (glomtable-cidi-frame-p match-cidi)
        2
        1))
    (let ((match-proto-depth (prototype-depth match-cidi query-cidi)))
      (if match-proto-depth
        (make-instance 'score
          :q-is-proto match-proto-depth)
        (let ((match-home-depth (home-depth match-cidi query-cidi)))
          (if match-home-depth
            (make-instance 'score
              :q-is-proto match-home-depth)
            (let ((query-proto-depth (prototype-depth query-cidi match-cidi)))
              (if query-proto-depth
                (make-instance 'score
                  :m-is-proto query-proto-depth)
                (let ((query-home-depth (home-depth query-cidi match-cidi)))
                  (if query-home-depth
                    (make-instance 'score
                      :m-is-proto query-home-depth)
                    (if (and (frame-prototype query-cidi)
                        (frame-prototype match-cidi)
                        (eq (frame-prototype query-cidi)
                          (frame-prototype match-cidi)))
                      (make-instance 'score :sibling 1)
                      (if (find-common-prototype query-cidi match-cidi)
                        (make-instance 'score :bad-match 1)
                        (make-instance 'score))))))))))))))))))

```

;;; **reindex-search-hash-table**

;;; *switch from storing matches as compounds (atemporal)*

;;; *to the compound's framer-frame (which contains its*

;;; *temporal info, which is used extensively later on)*

;;; *sort the matches by their score, and bind the hash-table*

;;; *to *hit-hash-table* for debugging purposes*

```

(defun reindex-search-hash-table (search-table)
  (maphash #'(lambda (compound-of-query compound-of-hit-assoc-list)
    (mapc #'(lambda (compound-score-with-components)

```

```

(let ((compound (first compound-score-with-components))
      (setf (first compound-score-with-components) (framer-frame compound))))
  compound-of-hit-assoc-list)
  (setf (gethash compound-of-query search-table) (sort compound-of-hit-assoc-list
    #'score> :key #'second)))
  search-table)
(setf *hit-hash-table* search-table))

```

;;; **convert-hit-hash-table-to-vector-of-vectors**

;;; create a vector of vectors of matches for the given list of compounds
 ;;; limit each vector of matches by the semantic threshold
 ;;; semantic threshold is used as an index into the ordered list of matches
 ;;; we make sure to take all matches after the index which have the same
 ;;; score as the match at the index.
 ;;; the elements of the vector of matches are lists whose first element is the
 ;;; score of the match, and the second element is the actual match

```

(defun convert-hit-hash-table-to-vector-of-vectors (compounds hit-hash-table semantic-threshold)
  (let* ((number-of-compounds (length compounds))
        (vector-of-vectors (make-array number-of-compounds)
          (index 0))
        (dolist (compound compounds)
          (let* ((value (gethash compound hit-hash-table))
                (number-of-matches (length value))
                (local-threshold semantic-threshold))
            (if (> semantic-threshold number-of-matches)
                (setf local-threshold number-of-matches)
                (do ((scores-at-threshold (nthcdr (1- semantic-threshold) value) (cdr scores-at-threshold)))
                    ((or (null scores-at-threshold)
                        (null (second scores-at-threshold))
                        (score> (second (car scores-at-threshold)) (second (second scores-at-threshold)))))
                    (incf local-threshold)))
                (let ((score-pair-vector (make-array local-threshold)))
                  (dotimes (i local-threshold)
                    (setf (aref score-pair-vector i) (list (second (car value)) (first (car value))))
                    (setf value (cdr value))))
                  (setf (aref vector-of-vectors index) score-pair-vector)
                  (incf index)))
                hit-hash-table)
            vector-of-vectors))

```

;;; **find-best-templates**

;;; takes an ordered heap with path objects and explores them until it finds a certain threshold of
 ;;; finished paths, or there are no paths to explore
 ;;; the ordered heap is a binary tree with nodes whose value is a list of all the elements with a score equal
 ;;; to the key of the node
 ;;; potentials is a list of the highest possible score theoretically achievable from starting at each
 ;;; step in the path (this score is calculated without regard to temporal consistency)

```
(defun find-best-templates (compounds vector-of-match-score-pair-vectors
                           temporal-constraints matches-threshold completeness-threshold)
  (let ((total-path-length (length vector-of-match-score-pair-vectors))
        (heap (insert-in-binary-tree nil (list (make-instance 'score) (make-instance 'score) nil 0)))
        (finished-templates nil)
        (potentials (calculate-potentials vector-of-match-score-pair-vectors)))
    (do ()
      ((or (>= (length finished-templates) matches-threshold)
           (null heap))
       (nreverse finished-templates))
      (let ((best-current-path (pop-best heap)))
        (multiple-value-setq (heap finished-templates)
          (expand-path best-current-path
                       vector-of-match-score-pair-vectors
                       temporal-constraints
                       heap
                       finished-templates
                       total-path-length
                       completeness-threshold
                       compounds
                       potentials))))))
```

;;; calculate-potentials

;;; creates a list of potentials

;;; the first element is the potential max score for the entire path

;;; the second is the potential max score for the sub path starting at step 2

;;; and so on

;;; the max score is calculated simply by summing the scores of the

;;; best matches for each step, without checking for temporal consistency

;;; we do check to see if no matches were found for a step, in which case

*;;; the potential for that step is the *null-score**

;;; this potential is obviously much higher than the score for the best consistent

;;; path, but as long as the potential is higher than the actual best score, the

;;; search algorithm will work correctly. The more accurate the potential

;;; is, though, the fewer paths which will need to be explored...

```
(defun calculate-potentials (vector-of-match-score-pair-vectors)
  (let ((potentials (list (make-instance 'score)))
        (index (1- (length vector-of-match-score-pair-vectors))))
    (do ()
      ((= index -1))
      (let ((score (make-instance 'score))
            (vector-of-matches (svref vector-of-match-score-pair-vectors index)))
        (add-scores score (first potentials))
        (add-scores score (if (and vector-of-matches
                                   (not (zerop (length vector-of-matches))))
                              (first (svref vector-of-matches 0))
                              *null-score*)))
        (decf index)
        (push score potentials))) potentials))
```

```

;;; make-path
;;; creates a path object from the given arguments
;;; this function used to calculate the next best index, and the
;;; potential of the next step, when the search algorithm only looked
;;; ahead one step. Now all this function does is return a
;;; path object with the potential which is retrieved from the
;;; list of potentials

```

```

(defun make-path (current-score path-list vector-of-match-score-pair-vectors constraints query-compounds
potentials)

```

```

  (if *debug-search*
      (format t "~%make:~A ~A" current-score (mapcar #'hit-index path-list)))
  (let* ((index-for-next-vector (length path-list))
         (number-of-vectors (length vector-of-match-score-pair-vectors)))
    (cond ((= index-for-next-vector number-of-vectors)
           (list current-score *null-score* (copy-path path-list) 0))
          (t
           (list (add-scores current-score (nth index-for-next-vector potentials))
                 (nth index-for-next-vector potentials)
                 (copy-path path-list)
                 0))))))

```

```

;;; expand-path
;;; expanding a path adds all possible matches at a current step that are temporally consistent and
;;; which don't match two elements of the query with the same annotation at the same time. The
;;; explored paths are inserted into the heap using their estimated total path score as a key.
;;; The path object has a key, which is the approximate score of the path when completed, a potential
;;; which is the estimated part of the key score, a path list, and a next-best index, which is the index
;;; for the best temporally consistent match in the next step.
;;; the path list is made up of steps, which have a temporal extent (by default, the extent of the match, but
;;; it may be clipped to satisfy the temporal constraints of the query) and an index into the vector of
;;; matches. If the index is -1, the step is empty
;;;
;;; when expanding, we first subtract out the potential from the key, which leaves the current score
;;; of the path in the key
;;; if the path is finished, we check to see the number of filled steps is greater than the
;;; completeness-threshold
;;; if so, we create a template from the path and add it to the finished templates list, unless it is equal
;;; to one of the already finished templates. We can distinguish between two types of equality, simple
;;; temporal extent equality, and semantic equivalence of segments, which checks to see if the segments
;;; subsume each other (which is possible when a match is found by backing off and leaving a step
;;; empty which would have further temporally constrained the result. Example: query for a temporal= b,
;;; find c temporal= d. we could also return just c, which might have a different extent from c temporal= d,
;;; but if we want diverse results, we should return an e temporal= f, where e and f are not as
;;; good matches)
;;;
;;; otherwise, we add all the paths with one additional step, checking temporal consistency

```

```

(defun expand-path (path vector-of-match-score-pair-vectors constraints heap finished-templates
                    max-path-length completeness-threshold query-compounds potentials)
  (if *debug-search*
      (format t "~%expand: ~A" (mapcar #'hit-index (third path))))
  (let* ((key (first path))
         (potential (second path))
         (path-list (third path))
         (next-best (fourth path))
         (index-into-vector-of-slots (length path-list)))
    (subtract-scores key potential)
    (cond ((= index-into-vector-of-slots max-path-length)
           (if (>= (count-if-not #'(lambda (step)
                                    (= (hit-index step) -1))
                               path-list)
               completeness-threshold)
             (let ((finished-template (make-template-from-path query-compounds
                                                              path-list
                                                              vector-of-match-score-pair-vectors
                                                              constraints)))
               (pushnew finished-template finished-templates
                        :test (if *diverse-results*
                                #'semantically-equivalent-sequences
                                #'equal-sequences)
                        :key #'movie-sequence))))
          (t
           (let* ((vector-expanding (svref vector-of-match-score-pair-vectors index-into-vector-of-slots))
                  (step-object (make-instance 'step :hit-index (1- next-best)))
                  (length-of-vector-expanding (length vector-expanding)))
             (dotimes (i (- length-of-vector-expanding next-best))
               (let ((new-path (nconc (copy-path path-list) (list step-object))))
                 (set-step-info step-object (1+ (hit-index step-object)) vector-expanding)
                 (when (and (temporally-consistent-path constraints new-path vector-of-match-score-pair-
                                                           vectors)
                           (not (same-overlapping-match-in-path
                                (second (svref vector-expanding (hit-index step-object)))
                                new-path
                                vector-of-match-score-pair-vectors
                                query-compounds)))
                   (let* ((score-for-last-step (first (svref vector-expanding (hit-index step-object))))
                          (score-for-path (add-scores (copy-instance key) score-for-last-step)))
                     (setf heap (insert-in-binary-tree heap (make-path score-for-path
                                                                        new-path
                                                                        vector-of-match-score-pair-vectors
                                                                        constraints
                                                                        query-compounds
                                                                        potentials))))))
                 (setf (hit-index step-object) -1)
                 (setf heap (insert-in-binary-tree heap (make-path key
                                                                    (nconc (copy-path path-list) (list step-object))
                                                                    vector-of-match-score-pair-vectors
                                                                    constraints))))))))))

```

```

                                query-compounds
                                potentials))))))
(values heap finished-templates))

;;; temporally-consistent-path
;;; checks to see if all of the temporal constraints are satisfied by the given path

(defun temporally-consistent-path (constraints path vector-of-match-score-pair-vectors)
  (dolist (constraint constraints t)
    (unless (constraint-satisfied-by-path constraint path vector-of-match-score-pair-vectors constraints)
      (return nil))))

;;; constraint-satisfied-by-path
;;; a constraint can be satisfied in several ways
;;; a constraint is considered satisfied if either of the matches is empty. This allows partial matches to be
;;; considered temporally consistent.
;;; if matches have been found for both query compounds, the function calculates the temporal relation
;;; between the two matches and compares it to the desired relation
;;; if the desired relation equals the calculated relation, all is well
;;; if no relation is found between the matches, this implies that the two matches are from different
;;; time lines. If the desired relation is a non-overlapping relation, we consider it satisfied.
;;;
;;; if the desired relation is a non-overlapping relation, it will match against any other non-overlapping
;;; relation
;;;
;;; the other cases can be summarized as two additional cases
;;; 1- the desired relation is some overlapping relation, and the calculated relation
;;; is an overlapping relation which can be clipped to become the desired relation.
;;; If this clipping occurs, the path is rechecked for temporal consistency
;;; the old temporal extent of the step is saved on the stack while we recheck the path
;;; if the path is inconsistent, we undo the change to the temporal extent of the step
;;; 2- the desired relation is starts or finishes and the calculated relation is equals. This is
;;; allowed (This is a case of relaxation)
;;;
;;; Ideally, the temporally consistent function would score the consistency of the path based on the
;;; desired type of matches, and would allow the clipping and relaxation only optionally. The current
;;; function is geared towards "constructing" the best possible match, not "finding" the best match in the
;;; archive.

(defun constraint-satisfied-by-path (constraint path vector-of-match-score-pair-vectors constraints)
  (let* ((step1 (nth (slot1 constraint) path))
         (step2 (nth (slot2 constraint) path))
         (temporal-relation (relation constraint)))
    (or (null step1)
        (= -1 (hit-index step1))
        (null step2)
        (= -1 (hit-index step2))
        (let* ((score-vector1 (svref vector-of-match-score-pair-vectors
                                     (slot1 constraint)))
               (score-vector2 (svref vector-of-match-score-pair-vectors
                                     (slot2 constraint))))
          (temporal-relation score-vector1 score-vector2))))))

```

```

        (slot2 constraint)))
(match1 (second (svref score-vector1 (hit-index step1))))
(match2 (second (svref score-vector2 (hit-index step2))))
(temporal-relation-between-matches (when (eq (media-time-line-frame match1)
      (media-time-line-frame match2))
    (direct-compute-temporal-relation (start-frame step1)
      (end-frame step1)
      (start-frame step2)
      (end-frame step2))))))
(cond ((eq temporal-relation temporal-relation-between-matches) t)
      ((null temporal-relation-between-matches)
       (dolist (el *non-overlapping-temporal-relations*)
         (when (eq el temporal-relation)
           (return t))))
      ((and (eq temporal-relation :=)
            (dolist (el *non-overlapping-temporal-relations* t)
              (when (eq el temporal-relation-between-matches)
                (return nil))))
       ;; trying to satisfy equals, and we have an some overlap
       (let* ((old-start-frame1 (start-frame step1))
              (old-start-frame2 (start-frame step2))
              (old-end-frame1 (end-frame step1))
              (old-end-frame2 (end-frame step2))
              (new-start-frame (max old-start-frame1 old-start-frame2))
              (new-end-frame (min old-end-frame1 old-end-frame2)))
         (setf (start-frame step1) new-start-frame
               (start-frame step2) new-start-frame
               (end-frame step1) new-end-frame
               (end-frame step2) new-end-frame))
         (if (temporally-consistent-path constraints path vector-of-match-score-pair-vectors)
             t
             (progn
              (setf (start-frame step1) old-start-frame1
                    (start-frame step2) old-start-frame2
                    (end-frame step1) old-end-frame1
                    (end-frame step2) old-end-frame2)
              nil))))
      ((and (or (eq temporal-relation :s)
                (eq temporal-relation :si)
                (eq temporal-relation :fi)
                (eq temporal-relation :f))
            (eq temporal-relation-between-matches :=)))
      ((or (and (eq temporal-relation :s)
                (or (eq temporal-relation-between-matches :oi)
                    (eq temporal-relation-between-matches :di)))
            (and (eq temporal-relation :si)
                 (or (eq temporal-relation-between-matches :o)
                     (eq temporal-relation-between-matches :d))))
          (let* ((old-start-frame1 (start-frame step1))
                 (old-start-frame2 (start-frame step2))
                 (new-start-frame (max old-start-frame1 old-start-frame2)))

```

```

(setf (start-frame step1) new-start-frame
      (start-frame step2) new-start-frame)
(if (temporally-consistent-path constraints path vector-of-match-score-pair-vectors)
    t
    (progn
      (setf (start-frame step1) old-start-frame1
            (start-frame step2) old-start-frame2)
      nil))))
((or (and (eq temporal-relation :f)
          (or (eq temporal-relation-between-matches :o)
              (eq temporal-relation-between-matches :di)))
     (and (eq temporal-relation :fi)
          (or (eq temporal-relation-between-matches :oi)
              (eq temporal-relation-between-matches :d))))
 (let* ((old-end-frame1 (end-frame step1))
        (old-end-frame2 (end-frame step2))
        (new-end-frame (min old-end-frame1 old-end-frame2)))
  (setf (end-frame step1) new-end-frame
        (end-frame step2) new-end-frame)
  (if (temporally-consistent-path constraints path vector-of-match-score-pair-vectors)
      t
      (progn
        (setf (end-frame step1) old-end-frame1
              (end-frame step2) old-end-frame2)
        nil))))
((and (dolist (el *non-overlapping-temporal-relations*)
      (when (eq el temporal-relation)
        (return t)))
     (dolist (el *non-overlapping-temporal-relations*)
      (when (eq el temporal-relation-between-matches)
        (return t))))))
(t
 nil))))))

```

;;; same-overlapping-match-in-path

;;; checks to see if a match is already used to represent something else at the same time
 ;;; loops through the steps, finds the match for the step, and checks to see if it's identical
 ;;; to the given match and if the queries for the two matches overlap.

```

(defun same-overlapping-match-in-path (match-component-frame path vector-of-match-score-pair-vectors
                                       query-compounds)
  (let* ((old-path-length (1- (length path)))
         (match-compound (find-compound-from-component-frame match-component-frame))
         (query-component-frame (framer-frame (nth old-path-length query-compounds)
                                              (index 0)))
         (dolist (step path)
           (unless (= index old-path-length)
             (let ((match-for-path-step (and (not (minusp (hit-index step)))
                                             (second (svref (svref vector-of-match-score-pair-vectors) index)
                                                           (hit-index step))))))

```

```

    (when (and match-for-path-step
              (eq match-compound
                  (find-compound-from-component-frame match-for-path-step))
              (overlap-p query-component-frame (framer-frame (first query-compounds))))
      (return t))
    (setf query-compounds (cdr query-compounds))
    (incf index))))))

```

;;; **score>**

;;; *calculates if score1 is greater than score2*

```

(defun score> (score1 score2)
  (declare (optimize (speed 3) (safety 0))
           (inline exact q-is-proto m-is-proto sibling bad-match))
  (let ((exact1 (exact score1))
        (exact2 (exact score2)))
    (declare (fixnum exact1 exact2))
    (if (/= exact1 exact2)
        (> exact1 exact2)
        (let ((q-is-proto1 (q-is-proto score1))
              (q-is-proto2 (q-is-proto score2)))
          (declare (ratio q-is-proto1 q-is-proto2))
          (if (/= q-is-proto1 q-is-proto2)
              (cond ((zerop q-is-proto1) nil)
                    ((zerop q-is-proto2) t)
                    (t
                     (< q-is-proto1 q-is-proto2)))
              (let ((m-is-proto1 (m-is-proto score1))
                    (m-is-proto2 (m-is-proto score2)))
                (declare (ratio m-is-proto1 m-is-proto2))
                (if (/= m-is-proto1 m-is-proto2)
                    (cond ((zerop m-is-proto1) nil)
                          ((zerop m-is-proto2) t)
                          (t
                           (< m-is-proto1 m-is-proto2)))
                    (let ((sibling1 (sibling score1))
                          (sibling2 (sibling score2)))
                      (declare (fixnum sibling1 sibling2))
                      (if (/= sibling1 sibling2)
                          (> sibling1 sibling2)
                          (let ((bad-match1 (bad-match score1))
                                (bad-match2 (bad-match score2)))
                            (declare (fixnum bad-match1 bad-match2))
                            (if (/= bad-match1 bad-match2)
                                (> bad-match1 bad-match2)
                                (> (temp-match score1) (temp-match score2))))))))))))))

```



Appendix C

Media Streams User Study Games

Media Streams User Study Games



We decided to introduce our subjects to Media Streams through a series of short games. Marc and Golan devised three different games which, we hoped, would be a fun way of teaching several of the basic skills needed to operate the system. Each game pitted our four two-person “teams” against each other in “races” to accomplish three specific tasks:

- **The Treasure Hunt Game.** Subjects were given a list of twenty Workshop icons (such as “toaster,” “fireplace,” and “to kiss”) and told to find as many of them as possible within fifteen minutes’ time. This game was the subjects’ first introduction to the Icon Workshop (a navigable, dictionary-like hierarchy of the system’s iconic descriptors) and bred the skills of navigating the Workshop and “reading” icons. The most successful teams were able to find 14 of the 20 required icons in the time allowed.
- **The Race Game.** Subjects were given a list of ten Compound icons (consisting of a total of 26 Workshop icons; an example was “inside a coffeehouse in San Francisco”) and told to construct as many of them as possible within fifteen minutes’ time. This game endeavored to broaden the subject’s ability to read and locate Workshop icons; it also, however, introduced the new skill of combining Workshop icons into Compound icons.
- **Actionary — the Glom Game.** This game required each two-person team to send a member (the “A” person) to the experimenter’s station. There, the “A” people saw one or two Glommed action icons on the experimenter’s screen. The “A” people were then asked to return to their team-mates (the “B” people) and attempt to convey to the “B” people — through action only — the meaning of the Glommed icons they saw. The “B” people were then required to re-construct (as quickly and as accurately as possible) the original Glommed icons at their own stations. The “A” people were instructed to call out to an experimenter when they felt that their “B” person had correctly re-constructed the Gloms; the first team to correctly produce the Glommed icons were declared to have won that round. Some of the Actionary tests that the subjects were given were: “chewing,” “spinning around while talking,” and “squatting; then, patting the floor.” This exercise allowed the continued development of the subjects’ icon-reading and icon-finding skills, while introducing them more deeply to the iconic Action hierarchy and the skills used in constructing Glommed icons on the Timeline. More importantly,

however, subjects gained an understanding of how they might actually annotate an action. They learned the critical skills of translating actions into graphic descriptions — without the help or hindrance of intermediating words — and creating Timeline annotations to convey the temporal relationships of these actions.

The Treasure Hunt Game

The object of this game is to find as many of the following unary Workshop icons in as short a time as possible. The first group to finish with the highest number of icons found, wins. You will have a total of fifteen minutes to play.

Before you start, drag the “All Icons” button into the Icon Space Query Bar, so that all the icons you find will be displayed.

toaster
to blink
South Africa
fireplace
left-hand region of the screen

bicycle
detective
to kiss
the 1960's
broadsword

to puff smoke
extreme close-up shot on person
beard
afternoon
rocking chair

tripod pan to the right
teapot
educational facility
dentist
to disperse (by a crowd)

The Race Game

The object of this game is to construct as many of the following Compound icons in as short a time as possible. The first group to finish with the highest number of icons made, wins. You will have a total of fifteen minutes to play.

Before you start, drag the “All Icons” button into the Icon Space Query Bar, so that all the icons you find will be displayed.



Actionary — the Glom Game

Each two-person team will send a member (the “A” person) to the experimenter’s station. There, the “A” people will see one or more Glommed icons on the experimenter’s screen. The “A” people will then return to the other member of their team (the “B” people) and attempt to portray, through action only, the meaning of the Glommed icon(s) they saw. The “B” people will attempt to re-construct the original Glommed icons at their own station, as quickly and as accurately as possible. When the “A” person feels that the “B” person has correctly re-constructed the Glommed icon(s), s/he should call out to the experimenter. If B’s reconstruction is incorrect, play will continue; otherwise, the first team to correctly produce the Glommed icon will win that round.

This exercise will be repeated four times, and the “A” and “B” roles will switch each time.

Experimenter’s Notes: Actionary Activities



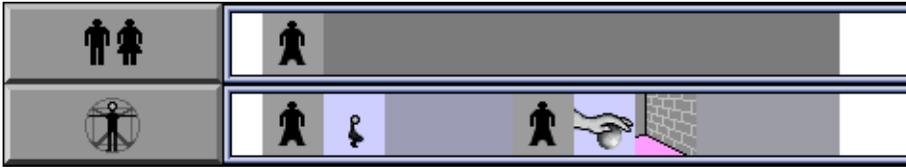
Demonstration: Walking while coughing; then, swiveling the pelvis



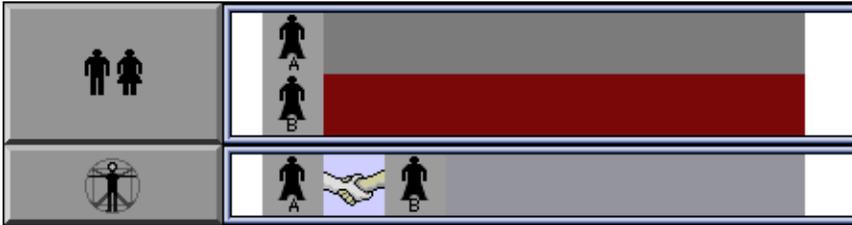
1. chewing



2. spinning around while talking



3. squatting; then, patting the floor



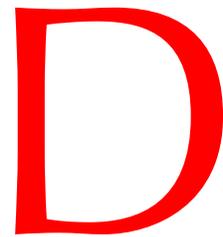
4. shaking hands



Appendix D

**Media Streams User Study
Exit Questionnaire**

Media Streams User Study Exit Questionnaire



Thanks for participating in our study. Your identity will remain confidential. You can answer any or all of the following questions; feel free to leave any question(s) blank.

Name: _____ **Age:** _____ **Sex:** _____

Attended Film School?
 Attended Art School?
 Studied Film Theory?
 How many hours of television do you watch per week?
 How many movies do you see per week?

Relevant Prior Experience	none							lots
Computer Experience	1	2	3	4	5	6	7	
Macintosh Experience	1	2	3	4	5	6	7	
Word Processing Software	1	2	3	4	5	6	7	
Image Editing Software	1	2	3	4	5	6	7	
Digital Video Editing Software	1	2	3	4	5	6	7	
Video Logging Software	1	2	3	4	5	6	7	
Experience in Repurposing Video, Film or Audio	1	2	3	4	5	6	7	
Video or Film Shooting	1	2	3	4	5	6	7	
Video or Film Editing (without a computer)	1	2	3	4	5	6	7	
Video or Film Logging (without a computer)	1	2	3	4	5	6	7	
Knowledge of Film Theory	1	2	3	4	5	6	7	
Prior Media Streams Knowledge	1	2	3	4	5	6	7	
Prior Media Streams Experience	1	2	3	4	5	6	7	

How easy or difficult was it to LEARN how to:

	easy						hard
Navigate the Movie	1	2	3	4	5	6	7
Navigate the Icon Workshop	1	2	3	4	5	6	7
Find a desired Icon in the Workshop	1	2	3	4	5	6	7
Read a unary Icon	1	2	3	4	5	6	7
Read Compound or Glommed Icons	1	2	3	4	5	6	7
Make a Compound Icon	1	2	3	4	5	6	7
Make a Glommed Icon	1	2	3	4	5	6	7
Make an Icon Query	1	2	3	4	5	6	7
Find an Icon in the results of a Query	1	2	3	4	5	6	7
Make an Annotation	1	2	3	4	5	6	7
Edit an Annotation	1	2	3	4	5	6	7
Translate what you saw into icons	1	2	3	4	5	6	7
Log with Media Streams	1	2	3	4	5	6	7

Please draw the Media Streams Learning Curve as you experienced it. Label your own axes.



How easy or difficult was it to DO:

	easy						hard
Navigate the Movie	1	2	3	4	5	6	7
Navigate the Icon Workshop	1	2	3	4	5	6	7
Find a desired Icon in the Workshop	1	2	3	4	5	6	7
Read a unary Icon	1	2	3	4	5	6	7
Read Compound or Glommed Icons	1	2	3	4	5	6	7
Make a Compound Icon	1	2	3	4	5	6	7
Make a Glommed Icon	1	2	3	4	5	6	7
Make an Icon Query	1	2	3	4	5	6	7
Find an Icon in the results of a Query	1	2	3	4	5	6	7
Make an Annotation	1	2	3	4	5	6	7
Edit an Annotation	1	2	3	4	5	6	7
Translate what you saw into icons	1	2	3	4	5	6	7
Log with Media Streams	1	2	3	4	5	6	7

Your Thoughts

What was easy?

What was hard?

What was fun?

What was frustrating?

What did you like?

What didn't you like?

How could you imagine using Media Streams in your own work?

How could you imagine changing the system?

How could you imagine changing the study?

What would you like to see happen with the research?

To whom would you recommend this system?

Do you have any other thoughts, comments, or suggestions?



Appendix E

**Media Streams User Study
Exit Questionnaire Results**

User Study Exit Questionnaire Results



New Users Numerical Results

- S1 Joshua
- S2 Raphael
- S3 Jane
- S4 Sarah
- S5 Erin
- S6 Vladimir
- S7 Sandra
- S8 Betsy
- A1 Nathan (alternate)

Background

	S1	S2	S3	S4	S5	S6	S7	S8	Total	Average
Attended Film School?	Y	Y	Y	N	N	N	N	N	3	0.38
Attended Art School?	N	Y	N	Y	N	N	Y	N	3	0.38
Studied Film Theory?	Y	Y	Y	N	N	N	Y	Y	5	0.63
How many hours of television do you watch per week?	2	15	7	0	10	13	10	15	72	9.00
How many movies do you see per week?	0.50	4	1	0.25	4.50	3	1	1	15.25	1.91

Relevant Prior Experience

Computer Experience	5	7	3	5	7	7	6	5	45.00	5.63
Macintosh Experience	5	7	4	5	7	7	6	2	43.00	5.38
Word Processing Software	5	7	4	7	7	7	6	6	49.00	6.13
Image Editing Software	3	7	2	6	1	4	5	4	32.00	4.00
Digital Video Editing Software	4	7	2	1	1	1	1	4	21.00	2.63
Video Logging Software	6	4	6	1	7	2	2	4	32.00	4.00
Experience in Repurposing Video, Film or Audio	6	7	1	1	4	3	2	3	27.00	3.38
Video or Film Shooting	7	7	7	1	6	4	4	4	40.00	5.00
Video or Film Editing (without a computer)	7	7	7	1	3	4	3	3	35.00	4.38
Video or Film Logging (without a computer)	7	5	7	1	7	4	2	2	35.00	4.38
Knowledge of Film Theory	5	5	6	1	2	2	3	3	27.00	3.38
Prior Media Streams Knowledge	3	4	5	4	1	1	1	1	20.00	2.50
Prior Media Streams Experience	1	1	1	2	1	1	1	1	9.00	1.13
Total	64	75	55	36	54	47	42	42	415.00	51.88
Average	4.92	5.77	4.23	2.77	4.15	3.62	3.23	3.23	31.92	3.99

How easy or difficult was it to LEARN how to:

Navigate the Movie	2	4	3	1	5	2	4	4	25.00	3.13
Navigate the Icon Workshop	5	3	6	1	6	3	4	4	32.00	4.00
Find a desired Icon in the Workshop	3	3	7	1	7	4	6	7	38.00	4.75
Read a unary Icon	2	2	4	1	6	5	6	5	31.00	3.88
Read Compound or Glommed Icons	2	3	5	2	5	4	5	5	31.00	3.88
Make a Compound Icon	6	1	5	1	3	3	3	4	26.00	3.25
Make a Glommed Icon	6	3	6	1	3	4	3	3	29.00	3.63
Make an Icon Query	6	5	6	2	4	2	3	1	29.00	3.63
Find an Icon in the results of a Query	6	5	6	1	4	2	5	2	31.00	3.88
Make an Annotation	2	1	3	1	3	3	5	2	20.00	2.50
Edit an Annotation	2	1	4	3	3	4	5	2	24.00	3.00
Translate what you saw into icons	2	4	7	3	7	4	5	5	37.00	4.63
Log with Media Streams	4	3	6	4	7	3	5	6	38.00	4.75
Total	48	38	68	22	63	43	59	50	391.00	48.88
Average	3.69	2.92	5.23	1.69	4.85	3.31	4.54	3.85	30.08	3.76

How easy or difficult was it to DO:

Navigate the Movie	2	2	1	1	2	2	4	3	17.00	2.13
Navigate the Icon Workshop	2	2	6	2	6	3	2	2	25.00	3.13
Find a desired Icon in the Workshop	5	4	6	3	6	4	5	7	40.00	5.00
Read a unary Icon	3	3	4	3	5	5	5	3	31.00	3.88
Read Compound or Glommed Icons	3	4	5	3	5	4	5	5	34.00	4.25
Make a Compound Icon	5	1	6	1	2	3	4	3	25.00	3.13
Make a Glommed Icon	5	2	6	1	2	4	4	3	27.00	3.38
Make an Icon Query	6	4	6	2	4	2	4	3	31.00	3.88
Find an Icon in the results of a Query	6	4	6	1	4	2	6	3	32.00	4.00
Make an Annotation	2	2	2	5	2	3	5	3	24.00	3.00
Edit an Annotation	2	2	2	5	2	4	5	3	25.00	3.13
Translate what you saw into icons	5	5	6	3	7	4	6	5	41.00	5.13
Log with Media Streams	4	2	6	5	7	3	5	4	36.00	4.50
Total	50	37	62	35	54	43	60	47	388.00	48.50
Average	3.85	2.85	4.77	2.69	4.15	3.31	4.62	3.62	29.85	3.73

Expert Users Numerical Results

Background

	Golan	Brian	Total	Average
Attended Film School?	N	N	0	0.00
Attended Art School?	Y	N	1	0.50
Studied Film Theory?	Y	Y	2	1.00
How many hours of television do you watch per week?	10	14	24	12.00
How many movies do you see per week?	3	3	6	3.00

Relevant Prior Experience

Computer Experience	5	7	12.00	6.00
Macintosh Experience	6	7	13.00	6.50
Word Processing Software	5	7	12.00	6.00
Image Editing Software	7	6	13.00	6.50
Digital Video Editing Software	4	5	9.00	4.50
Video Logging Software	6	7	13.00	6.50
Experience in Repurposing Video, Film or Audio	7	3	10.00	5.00
Video or Film Shooting	3	2	5.00	2.50
Video or Film Editing (without a computer)	4	1	5.00	2.50
Video or Film Logging (without a computer)	1	1	2.00	1.00
Knowledge of Film Theory	5	3	8.00	4.00
Prior Media Streams Knowledge	7	7	14.00	7.00
Prior Media Streams Experience	7	7	14.00	7.00
Total	67	63	130.00	65.00
Average	5.15	4.85	10.00	5.00

How easy or difficult was it to LEARN how to:

Navigate the Movie			0.00	0.00
Navigate the Icon Workshop			0.00	0.00
Find a desired Icon in the Workshop			0.00	0.00
Read a unary Icon			0.00	0.00
Read Compound or Glommed Icons			0.00	0.00
Make a Compound Icon			0.00	0.00
Make a Glommed Icon			0.00	0.00
Make an Icon Query			0.00	0.00
Find an Icon in the results of a Query			0.00	0.00
Make an Annotation			0.00	0.00
Edit an Annotation			0.00	0.00
Translate what you saw into icons			0.00	0.00
Log with Media Streams			0.00	0.00
Total	0	0	0.00	0.00
Average	0.00	0.00	0.00	0.00

How easy or difficult was it to DO:

Navigate the Movie	3	2	5.00	2.50
Navigate the Icon Workshop	2	2	4.00	2.00
Find a desired Icon in the Workshop	3	3	6.00	3.00
Read a unary Icon	2	1	3.00	1.50
Read Compound or Glommed Icons	1	1	2.00	1.00
Make a Compound Icon	1	2	3.00	1.50
Make a Glommed Icon	3	3	6.00	3.00
Make an Icon Query	2	4	6.00	3.00
Find an Icon in the results of a Query	2	4	6.00	3.00
Make an Annotation	1	1	2.00	1.00
Edit an Annotation	3	1	4.00	2.00
Translate what you saw into icons	2	4	6.00	3.00
Log with Media Streams	2	3	5.00	2.50
Total	27	31	58.00	29.00
Average	2.08	2.38	4.46	2.23

New Users and Expert Users Comments

What was easy?

Joshua: Moving around, dropping things where I wanted them.

Raphael: After learning, it was easy to navigate. The interface is very clear — finding the right icons is hard but [the] general idea is on track.

Jane: It's frustrating. Easy concepts like: he's walking up the stairs of an office building, past big lion sculptures, and entering the building through a door. I accomplished: he's walking, there are stairs, he goes into the doorway of an office building.

Sarah: Understanding the interface, the concept behind the work and simple manipulations... I *should* say *physical* manipulations.

Erin: Using the Mac. Traversing the icon hierarchy. Dropping icons into the Timeline.

Vladimir: To navigate the Movie. To use already-created icons.

Sandra: Not exactly easy because the task isn't an easy one but the program was pretty easy for me to use at these levels of use.

Betsy: Moving an icon onto the stream.

Nathan: It wasn't easy. But it was fascinating. I didn't fully participate, so my thoughts are based on a very partial trial experience. There are definitely hurdles to overcome: technically with the interface, psychologically with thinking iconically (in an imagistic grammar); practically learning a significant amount of structure and (iconic) terminology.

Golan: Making a unary compound and annotation. Scrubbing with the red rectangles.

Brian: Logging visually clear video, and simple translational actions.

What was hard?

Joshua: Finding ways to express certain (more complex) "phrases."

Raphael: Dealing with slow-downs — I know it is the Alpha, but still... Adjusting to quirks — lock-ups, slow-downs. Translating it all into icons — why not a text/icon hybrid? or a text search thing?

Jane: I couldn't figure out how to express a lot of what I wanted to say. The same walking-into-the-building shot has a reverse angle — we see Ray from behind and then from in front. I had no idea how to do that with icons.

Sarah: Conceptual manipulations (understanding on what you should spend your logging time, what information was irrelevant, or simply understanding what a film-language consists of) were much more difficult for me.

Erin: Finding specific things in the icon hierarchy. Reading the icon hierarchy. Remembering/finding/switching modes. Constructing Glommed icons to represent human and object actions.

Vladimir: initially to construct complicated (glommed) icons. To find some unary icons without the "help" function on.

Sandra: Some things were hard, like getting the relationship between the cursor (with images attached) and the Timeline.

Betsy: Thinking through the logic of finding an icon... how would I find/make "photographing papers." Distinguishing "places" from "objects" from "object actions" → where did this "boat" come from? Changing what I'd already done. Couldn't easily replace an icon, though sometimes the editor did this for me.

Nathan: I found it hard to get an overall, general log of the clip I worked with (day 1), and became semi-lost in the details, missing the forest for the trees. One small scene can expand to a large logging task given the capabilities

of the system, which is excellent but potentially dangerous in terms of **completing** logging tasks.

Golan: Working with one hand tied behind my back: not having access to certain of the system's functionalities, such as the cotemporaneous linker.

Brian: Describing complex compound actions; describing ambiguous situations.

What was fun?

Joshua: Generally, logging was fun. Sort-of like a medium-hard math problem that you know you can solve. I like organizing things — this was beautiful in that respect. Making pure order out of chaos.

Raphael: The surreal poetry element is fun — the quiriness of the grammar is fun also — sort of stilted in a way I like — results of misinterpretation when translating.

Jane: Some of it is fun. Finally being able to say: a man named Ray is wearing a lab coat. Its like a syntax game with images. So that's kinda fun. But when you can't say: he's taking pictures with a little spy camera — that's frustrating.

Sarah: Moving around in the icon hierarchies. The animicons. The process (in very general terms).

Erin: The games you set up for learning. Guessing what an icon was supposed to represent.

Vladimir: Everything. In particular, once you know how to do it, to construct glommed icons.

Sandra: Yes, I thought so, but again, in this context. As a full-time job, no, it would not be fun. The icons were fun to use and moving images and having them "drop" onto the right place was fun.

Betsy: Working in a visual way with an array of icons.

Nathan: The "smartness" of the system is remarkable and created enjoyment: Especially in the simplicity of drag&drop; the simultaneity of streams; the multiplicity of information-zones put in relation to each other; the visual design of the interface (elegant, expandable, re-arrangeable, slick, suave...); and equally importantly the *fun* of anticipating using Media Streams to log films I'm discussing in my dissertation.

Golan: Easy footage that had reusable icons.

Brian: Creating annotations. Using gloms.

What was frustrating?

Joshua: Just the sheer amount and complexity of information to be logged. Also, wasting time by watching the movie, then finding myself out of place and needing to navigate back.

Raphael: Same as what was hard... Also, expressing the level of detail in icons as one can in words is tough... makes one feel like an infant... know what you want to say but can't get the words out.

Jane: Being constrained by the language of your icons. Not knowing how to make things happen well enough (i.e. making compounds and gloms, etc.).

Sarah: The inability to find an icon you know you'd seen before. The fact that the bar doesn't move as you scrub with the thumbnails, but only jumps to that point when you stop. General slowness or jumpiness — but I suppose this is to be expected.

Erin: Manipulating icons for logging activities on screen. Deciding on what to log: the Icon space *invites* very detailed logging. Understanding all the "doodads" on the interface (bars: grabbing, moving, etc.; single or double-clicking, etc.). Moving the Movie window around.

Vladimir: Bugs. To get used to grab icons to move. When you scroll through bigger sections to use the mouse to drag the timeline.

Sandra: No, not in this context, but it would be if I had to "do it right" or "like everybody else," I think.

Betsy: Not finding icons — like "wearing"; the people actions were particularly frustrating — especially the transitive verbs like wearing as opposed to "abstract arm movements."

Nathan: A frustration reminiscent of learning a new language — that is, a pleasurable frustration because the results are cumulative, but also difficult at first because you feel that you can't "say that much" initially. The workshop incorporated games, which was excellent, and Media Streams resembles an (elaborate/intellectual) game, i.e., it's seductive. The architecture and philosophy built into it are compelling.

Golan: Crufty footage with no re-usable icons (difficult descriptions); sticky computers; bugs with redraw; trying to find an icon in the result of a query; not being able to grab an element of a compound; not being able to scrub past the edge of the monitor; not being able to jump to the end of a frame-region (annotation); trying to grab the select-bar.

Brian: Maintaining consistency of actions and existence of objects at the frame level. If an object isn't visible for 10 frames, should it be removed? This is time-consuming.

What did you like?

Joshua: The general layout. The relative richness of descriptors.

Raphael: I like the level of detail... the camera stuff — cinematography — is very cool; good way of standardizing a lot of info. Also, its "post-symbolic communication" feel is cool.

Jane: Moving through digitized movies on the screen.

Sarah: The feel of the system; the liveliness added to a task that could be drudgery.

Erin: Visually pleasing. Iconic language seems to be [a] move in correct direction for fast analysis of film/video clips.

Vladimir: Fun icons. Good animations of some motion. Pseudo-intelligence of the system in some situations.

Sandra: I like the way you've worked out a visual language and the look of the program is very nice. Also, the tool is needed and I like that it is happening.

Betsy: Thinking of what might be re-used, how someone might pose a query, then figuring out how to communicate symbolically with them. I like the smeared scene stream [videogram] — it helped me orient to the people and actions.

Brian: The intelligence of the interface. It tried to do the right thing, and it did a pretty good job.

What didn't you like?

Joshua: several icons seemed to just "not be there" or to be difficult to find. I wanted more cross-referencing. Speed was sometimes slow.

Raphael: Logging ugly footage, old random B&W stuff. I think the system works best with certain kinds of materials — it is tough looking at a dark QuickTime movie & extracting stuff.

Jane: The constraint of explaining moving images with icons.

Sarah: That I "couldn't use" icons from differing superordinates (e.g. "institutional building" with "inside") as a character position, not only as a scene location: I wanted to use icons in slots they weren't intended for.

Erin: Invites a specificity of logging I'm not comfortable with. Too many streams that need to be considered. I didn't feel it was appropriate for me to choose streams I was interested in and ignore the rest.

Vladimir: Need more screen space. Slow response to some actions.

Sandra: The open-endedness of the degree of detail. The layout when opening a new stream should be customized for personal use.

Betsy: See frustrations.

Brian: The latency of many actions.

How could you imagine using MS in your own work?

Joshua: By annotating pan and track movement in detail, then using this info to find shots to make an abstract music/video motion piece.

Raphael: As another layer of info — would be cool to encode MS data as a track... logging random stuff is a bit tedious — I would be more interested in my own work with my own icons — much more personalized — with a way of transferring my glossary & pref's to others.

Jane: To retrieve. To say I want shots of one man walking into a big office building. Or an old shot, B&W, of waves crashing against the shore.

Sarah: it's a stretch, but in art, the icons could be a blast. e.g. Hamlet in iconese.

Erin: Because I'm more interested in content of audio stream, I'm not as interested in video stream. However, at very high level pass, might help point to the appropriate visual scenes (e.g. "using telephone")

Vladimir: I'm not sure yet.

Sandra: I can imagine studying the use of visual language and how the meaning of visual artifacts comes to be shared.

Betsy: Creating a smaller set of more custom as opposed to generic icons. My video bases are more constrained... and detailed.

How could you imagine changing the system?

Joshua: Being able to type in a word to see if such an object or action exists.

Raphael: I would love to use it in reverse, grabbing icons then searching for matches. Also, taking certain elements from found tracks, let's say camera motion, & applying it to different footage — I guess, if each element could be isolated & used with other stuff it would be cool.

Sarah: Big change: using networks or radial structures for icon access rather than just hierarchies, or/and more work with cross-indexing of semantic fields.

Vladimir: In the Timeline to see definitely the lines for which I have dropped annotations so that I can delete them when no longer valid. I do not need to see all the lines that I have not logged about.

Sandra:

- making the streams thinner and the icons smaller.
- Being able to work in the icon space from the icons on the Timeline screen.
- Customizing the first timeline screen.
- Having more than 3 icons/glom.
- I'm afraid all the clicking all day long will be difficult on our hands.
- More group actions?

Betsy: Instead of lines, regions. Booleans in query language (i.e. "not")

Nathan:

- Lookup function, i.e. a dictionary within the system would be helpful (accessible through Find command).
- Zoom in/out, (but problem of reduced/illegible icons develops)
- Selective use of language in streams may or may not work but I would encourage users to name their icons for self-reference & greater expressivity.
- A workspace screen to play with a spatial, not necessarily chronological representation of a scene. One aspect of this could be arranging the relevant icons (like compound & glommed ones) in a space reserved for a specific clip or scene.
- Delete function for icons not serving purpose anymore in query space — select some with shift key & use mouse-shift to highlight and cut...Also keyboard return function to create more space for current accumulation of icons.
- Space, or another screen, to take notes (using written language) related to current logging task —> a "translation" space, so to speak.
- The amount of icons and the arrangement of hierarchies are both excellent. I would encourage more cross-referencing but this is difficult and might be confusing if *between* categories. It's better within a category, (e.g. character actions, etc.) obviously.

How do the search capabilities work? Can you ask for every driving scene across *lots* of digital footage that has been logged? How are databases of logged stuff to be handled/coordinated? Will digitized films be available en masse in our lifetime?

Brian: Lots of ways. Grabbing components. Editing compounds in place. Pasting components into existing compounds. "Macro" compounds for complex actions.

How could you imagine changing the study?

Joshua: More time early on for unhurried learning and question-asking

Raphael: The games were an excellent way of getting acquainted...some of the footage was very complicated for first-time loggers — more strategy tips would be cool also. An example of "perfect." Examples of logging styles would be good.

Jane: I was bad and didn't study the manual. I suggest more training time.

Sarah: A progression from more elemental footage (fewer objects, simpler actions, high clarity) to more complex footage during the training and logging. This may confuse self-perceived learning curves, but will allay some initial frustration.

Erin: Less difficult film clip for first logging session. 2.5 days is a MAJOR commitment although you haven't got a choice. I had troubles switching in/out of logging from time at work.

Vladimir: Perhaps add session when people actually annotate the whole sequence and compare times and reaction with respect to more traditional annotation methods.

Nathan: I didn't participate enough to say, but I would have appreciated more background on development, goals, history of the project; and more discussion of usefulness/adaptability to various media undertakings.

Brian: Changing the dependent part of the survey to make it more like real life.

What would you like to see happen with the research?

Raphael: I am excited to see it as a product — if it could be made faster, easier & smarter it would have a much wider audience. Why not build it into a camcorder? An auto-annotate button would be cool.

Jane: I think a universal logging system is a great idea. I'm wondering if it can be done with only icons. Maybe you need a few verbs to tie things together.

Sarah: I think it would be interesting to look into the possible ethical problems related to the decontextualization of certain footage in the light of its possible uses in repurposing. Should enabling unethical decisions be a concern?

Erin: Use it to redesign system, identify appropriate users, consider training & certification process.

Vladimir: A product.

Betsy: Products!

Brian: Turned into a product.

To whom would you recommend this system?

Joshua: Archivists, others involved in re-purposing video.

Raphael: Anyone making video or film — would be cool if it came as an extra information track on everything.

Jane: Stock footage houses.

Erin: Film stock companies.

Vladimir: Loggers for re-use of footage.

Betsy: Film/video houses.

Nathan:

- Film scholars in Paris.
- Home video and hacker-experimental type enthusiasts.
- Archivists of audio/visual materials.
- Giordano Bruno.

Brian: Archivists.

Do you have any other thoughts, comments, etc.?

Joshua: I can't see using this for normal logging at this point. Simple scribbles and keystrokes contain lots of personalized meaning which is necessary and sufficient. This process is a de-personalizing of the info, which makes it less valuable for "everyday" use. When I re-purpose video, the content's dialogue is crucial — this is outside of that realm, so I would probably just go on saving clips as I do.

Raphael: I guess I am most interested in working with logged footage (more than logging). Logging is a bit tedious. It feels good to log something well. Certain stuff seems more "loggable".

Sarah: It'd be great if you could just say "dog on couch," "man wearing lab coat" etc. You should have the little eyes that follow your cursor — dependent upon use of screens. For some reason I'd prefer working on a horizontal screen with this system, but not for watching the movie.



Appendix F

**Transcript of the Media Streams
User Study Wrap-Up Discussion**

Transcript of the User Study Discussion



- Marc:** Are there any comments?
- Sandra:** My hand hurts.
- Betsy:** From the repetitive mouse movements.
- Nathan:** You need a bigger mouse pad.
- Marc:** Because there's so much screen real estate, and you have to make wide movements?
- Everyone :** Yeah
- Joshua:** But I can see why you'd want a huge screen for this.
- Erin:** In fact I wanted to have a third one, I kept wanting to have the movie in another screen, because I was tired of having to move it all over the place. Like, excuse me, you're in the way, etc.
- Marc:** So what would be the optimal screen configuration, let's say if money were no object.
- Nathan:** Four times the 21-inch.
- Erin:** Either that or seeing the movie someplace else, like in a goggle or something, something you could bring or swing in and out horizontally.
- Marc:** Like virtual-vision glasses?
- Raphael, Joshua, Nathan:** *Naaah.*
- Sandra:** Or, it might be nice to have it here, below the main monitor, on a little viewer, so your eyes could just look up and down.
- Marc:** So how big would you want the movie to be?
- Betsy:** It would depend on how much you want logged. if you wanted to log more objects, then it needs to be bigger so you can see more objects.

Joshua: Not too much bigger.

Vladimir: This size was fine.

Erin: I like it smaller because it wasn't much in the way, and easier to move around.

...

Vladimir: One thing is I'd like to have the streams close up when I'm not looking at them. when you drop something in, to open up, since I don't look at it much anyway when I'm not logging it.

Marc: So you'd want it to shrink to what you have and when you drop something in to open up?

Vladimir: Yeah.

Brian: It was doing that today. I fixed it so that if you dropped it in today it would open up. The hard part is that, due to limited screen real estate, if it shows something you just added it may hide something you really wanted to see still.

Sandra: But the icons and the streams could be a little bit smaller, I thought.

Vladimir: They could adjust, so that if you only had a few streams they'd be wider, but if you had too many they'd get smaller.

Marc: Would you like control over the size then?

Vladimir: You don't have to have explicit control, just control dependent on how many streams are there. They get smaller and smaller as more streams are put in.

Marc: The one you're working on could be thicker than the others, like a fish-eye lens.

Brian: If you automatically scale the icons it might be difficult to recognize what they are.

Joshua: Yeah.

Sandra: Well you would have to have a lower limit.

Marc: How small do you think you could go and still use it? To give you a sense, these icons were 32 by 32 pixels.

Erin: I wouldn't go any smaller, at least until I got used to what they meant.

[General agreement]

Golan: Let me actually ask a few questions about the icons:

1. How sensible did you feel it was to log with icons?
2. What did you think of the organization of the icons, how sensible was it?
3. How readable was a given icon?

Erin: I'll pick the middle. The ontology of icons was more and more understandable as I understood the ontology as I went on, which is true of all ontologies; therefore you can do whatever you want, and eventually I would say it's natural, but at first I was like, this is braindamaged, for the first day or so. So eventually, once you force me to work with the set.

Marc: How long do you think it would take?

Erin: I think it would take another week, and then I could find everything I'd need to find.

Golan: Well, you said it was braindamaged, but then you learned it. To what extent do you think it really was braindamaged, versus the extent to which it really should be organized that way?

Erin: Golan, you don't ask that question, because ontological construction is something which is impossible to do right, or wrong, from my perspective — you do it for some purpose. And eventually a person who is working with your ontological distinctions will learn your purpose. Hopefully. You can do almost anything to a human being ontologically.

- Jane:** Yeah I think some of them, yeah I'd learn it, but I'd always hate it. Like that a phone is a media device and that it's an audio media device — to me there should be a Communications device icon that cascades, with fax machine, phone, and then if you want just pieces of paper, that's another media device, like a written media device.
- Marc:** What's the difference between a media device and a communication device?
- Jane:** See media to me means television, radio, media. It doesn't mean loose leaf.
- Erin:** I thought that was a very good idea, from a generalist perspective, the media device including paper. But I understand Jane's point — it's in use that it gets cumbersome. And I think that requires the ability of a person who has a particular vision of what a media device is, who says, "I want to redefine this because your philosophical use of media is different from my in-practice use of media", and that's why you eventually get used to the philosophical structure.
- Marc:** I want to get back to the learnability of the icon hierarchy. Erin said "in a week, you'd know how to find anything." How do the rest of you feel?
- Jane:** For certain things like phone or book, yeah, now I know where to go to find those, but for anything like a verb, uh, "he's taking a picture".. I still don't know how to do that. I've got Him and I've got a Camera, but I was really frustrated not being able to construct what was happening.
- Betsy:** The *people-object* relationships.
- Vladimir:** But that's the advantage actually of having pictures because also the interpretation after that is up to the person who's reading it. So I said well "taking a picture": he's using his hands, there is a camera, so there is him, there's his hands and a camera, so when we look at this, it's clear that he's taking a picture.. or at least that he's holding a camera... especially when there is the motion.

Jane: But I want to say, he's "sneaking around, he's got a little spy camera, he's like looking through the blinds" .. I didn't get any of that.

Betsy: But she's our filmmaker here, so she wants a whole level of actions because those are what's important to her, and I think that's worth thinking about.

Marc: Let's explore this for a second... One of the questions in Media Streams is logging for reusability. What I hear Vladimir saying, which is of course why we're all smiling, because this is part of the underlying way we think about the system, — but which may in fact not suit the needs that Jane's talking about, is that if you're going to find a piece of footage to use in a different way than it was originally intended, to make a new sequence, what's the representation to use? and Vladimir's saying, taking a picture is "holding a camera, he clicks on it, the camera's pointing in a certain direction..." and Jane's saying, "he's taking a picture, he's sneaking around —"

Jane: There's no "clicks on it". You threw that in there. There's no icon for that, is there?

Golan: There's an icon for "pressing." You could construct —

Jane (skeptical): you could "press the camera"?

Golan: You could press a button, which was on the camera.

Jane (sarcastic): You could press a button?

[Subjects laugh]

Joshua: This is kind of separate, but I really wanted to be able to just type in a word sometimes. I think icons are perfect for, I find it really easy to use for, the screen position of something, for what was going on with the camera, I don't think you could do that any other way, but sometimes its not worth the effort if you have to go eight levels deep into something, that's going to be more time than just typing in 5 letters.

Sandra: Like to find a book, I mean you know the book is in there and you want to just grab it, so and having to go in-in-in was a pain in the ass.

Joshua: And even when I had other people's stuff to work with, I don't know why but I didn't find myself using it as much as I might have, because then there was this extra step of, OK now I filter it so then it becomes easier to find.

Erin: What's interesting about what you just said is how easy it was to talk about the construction of the film. And that's because I think that's a very standard way of looking at the medium. Whereas all the objects, all the actions, if I sat down and I thought about if I was interested in logging this for X purpose, I could do that if I got rid of about 7 or 10 streams — I don't really want to work with all these others. And I realized that if I wanted to work for Y purpose, I would want to take maybe some of the same streams but maybe reconfigure the ontological assumptions underneath them. But I don't think that's true of the camera position because that's a very well worked out vocabulary, well worked-out descriptions -- when you look at a piece of a shot, that's something that if you go to film school, that's something you learn. That's not true when of when you look at someone picking up a telephone: some people will say he's picking up a telephone and have all sorts of description about the telephone, versus the picking up of the telephone, versus what else is on the table because he had to make a decision about which telephone, or what was the telephone. That's why I say the ontological underpinnings are both at one point for me braindamaged, and at another point natural. It's sort of, what do you give me to work with. Right now it's alot easier to describe the technology of shotmaking.

Vladimir *(to Marc):* Who do you see as users of the system, eventually?

Marc: The first users would be stock-footage archivists, people who are describing footage for re-use. It sort of relates to the purpose issue. The purpose of the description is to describe things so that they can be re-purposed. In other words, its not saying for a particular domain; instead, it's saying how can you come up with a description of the

footage such that it could be used in different ways than it originally was. And the people whose job that is now, are the people who work in stock footage libraries. So the system in its current incarnation is intended for that type of user.

Golan: How helpful was it to have other people's icons?

Sarah: I personally found it kind of disturbing because — well I don't know if it just has to do with learning the ontology like she was saying — but I would look for "someone has a hat on their head" and someone would have put in "someone has a hat on their hair," and I was like, should I use this or should I just make another one. There were a lot of things like that where they were there but then there was this issue of whether I should use them or not.

Joshua: I used those as a shortcut by bringing them up to the workshop, and you'd get to something near what you wanted.

Sarah: That's true.

Vladimir: I found it very useful actually because you could find pretty much a lot of the icons that you actually need: Just bring them up there instead of going and searching for them through all the hierarchies, and then use them to build your own. I also found it useful in making a first pass and describing the main events. Then later, if I had time I could go back in and just add stuff.

...
[Everyone agrees that having multiple palettes would have been useful, and they wished they had been told about that feature. Betsy suggests user-defineable palettes.]

...
Erin: One problem is that it's too easy to inadvertently "and" a query, particularly when the idea of throwing something away — I didn't quite get it over there and then all of a sudden I got this anded thing. I had to be so careful or else I got these anded categories.

Marc: Did people find it hard to throw things away?

Betsy: Sometimes when I was dragging a query off it would go onto the timeline.

Erin: You really had to drag things *down*.

Brian: There's not enough space on the desktop.

Marc: Would people prefer a trash can?

Jane: I thought it was easy to drag things down.

Sandra: I thought it was okay the way it was.

Betsy: Even just an undo would take care of the problem.

Brian: Uh, undo is currently only supported for actions on the timeline.

Sandra: One thing is that I would have compound icons in the palette that I wanted just one of the things in it.

Sarah: It would be really nice to select one element of a compound on the timeline and replace it. It would also be nice to drag a glommed icon up to the workshop and be able to inspect the first, second or third element, not just always the first.

Erin: Also I wanted to be able to do that on the timeline itself. Step through glom variants on the timeline rather than hunting through them in the palette. change a general purpose guy in a timeline glom to say, excuse me this is actually "Rock."

[In such a case, everyone kept trying to drag "Rock" onto the Time Line glom to replace the incorrect character.]

Erin: The Rock Hudson footage would be a good test case for that because you could just plop different characters into the same gloms for "waving".

Raphael: You could have cascading menus that you can just drag down rather than clicking each icon -- in place, on the timeline, recreating part of the workshop.

...

- Betsy:** Once I found out the balloon help was a toggle, I would turn it on and off. Most of the time I left it off because it stands in the way -- you can't see the icons when balloon help was on. So I would use it kind of as for confirmation.
- Erin:** Balloon help is a pain because it interacts with everything on the screen.
- Sandra:** Rather than have balloon help follow your mouse around everywhere, have it only show up while you held down a key.
- Vladimir:** A PC Mouse with two buttons, one for help.
- Sarah:** I found myself double-clicking to open in the workshop. Single-clicking requires unlearning.
- Joshua:** I got used to that. I liked it because it was faster to single click.
- Erin:** Other things in the system require double-clicking, don't they?
- Marc:** Was it worth unlearning it?
- Sandra:** I did double-click a few times but it was alright to have to unlearn that.
- Nathan:** is it possible to click and hold down and see the path? That's really useful.
- Jane:** On the topic of bubble help — I think once you've found something a few times you recognize it and you don't need the bubble help. But with things like those actions, I think it would take a long time to really know what the different ones were supposed to mean and it might be good just to have permanent words, 'cause that bubble thing is much longer than it needs to be, it only needs to be like three words long.
- Marc:** Why for the actions?
- Jane:** Because it wasn't intuitive to me most of the time.

Golan: Is that because the icons weren't clear?

Jane: Y'know like when I was trying to say "putting the book down on the table" and Marc said well it has to be "replace"... there's no "putting" or "let go."

Nathan: Did you have trouble assigning the word to individual icons with balloon help?

Brian: You can title the actions but right now it's disabled.

Jane: You mean make it up yourself?

Raphael: Yeah you'd just take hand action and call it whatever you want, like "grab."

Marc: I'm wondering about the model of what you think you were doing when you're logging actions.. it seems you're saying you're trying to find the right word, and the icon is showing you this action. Was the action that the icon was showing you not what you were looking for?

Jane: Yeah. He wasn't "replacing" the book, like dropping it, he was "putting" it down.

Sandra: I guess yeah, your hands are different.

Joshua: It's incredibly complex.

Betsy: I think in term of reuse, when I see movies that people have made from one-second clips, its those hand or body-object actions that you want, and those are the hard things... I didn't log very many of them, in retrospect.

Sandra: The problem was the word that you *did* give them. You could imagine that this would be "put", but if it says "replace" you wouldn't use it.

Marc: What if there were no words for the actions?

Sandra: That would be better almost

[several other users, Nathan, Erin say this.]

Erin: That's actually what I did, I never used balloon help for any of the actions and I still found it really difficult to figure out where the actions were — whether it was an object or a person or a displacement in x and y, it had so many places to figure out what I wanted to do.

Marc: It's clearly a very different way of thinking about what an action is than we're used to, and Betsy's point about that may be how you'd want to do it for repurposing .. I'd like to explore that for a little bit: A) do you think that's a good way, especially the film folks, of thinking about the actions for re-using, and B) is that a very difficult way to think about it in the system, is it very hard to log those sorts of things in the system?

Joshua: As far as repurposing, which I know something about, I actually know personally — maybe I'm just not thinking far enough into the future — but I can't personally think of making , as something I'd want to do , making something happen made up in the explicit way you're talking about, just made up of random other pieces. They're important because they have cultural significance attached to them, and it matters whose hand it is, and it matters where it came from, to me. And I don't start something with an idea, that well I want this to happen, I'm going to find the footage to do it; it's almost the reverse: here's the footage that I've seen, these things are memorable, I've kept them, I'm going to go and reuse them. I'm still wrestling with that because one way that I thought would be really interesting is having all these truck and pan motions and stuff, that would be great to use because I love the idea of decontextualizing that .. I think you could do something really fun with assigning different values of [e]motions to different kind of sounds.

Jane: But just think, if you were trying to do a montage and you've got someone diving off of a diving board .. so you want all the shots of people jumping off a diving board and then you re gonna make it into one big dive.

Joshua: See but I don't think that way. I mean I can see how I would, and how it would be useful.

Marc: Jane, could you elaborate?

Jane: Well, I could see where you could use a system like this for something like that. You could just put in an icon, and maybe you don't care if it's a man or a woman who's diving. I just don't want to be the one to log it, because I don't know where I'm going to find "diving board" and "jumping."

[everyone laughs]

Marc: If there were a stock footage house with expert users logging this stuff, would you like to have access then, as a filmmaker, to that footage?

Jane: Sure.

...

Marc: How long do you think it would take to become a total whiz on the system, and how fast could you log once you were?

Joshua: Two or three weeks of days like this.

Raphael: I would want to personalize it, to make my own icons and organize them myself, but that's a big problem if you want to share it with a lot of people, then how do you translate. Maybe you could make your own glossary.

Erin: I think it's a divisible process where different people could log different streams.

...

Sandra: I wonder maybe if I were French I would want the compound icons to be ordered differently...I found myself looking at the compound icons as a meaningful unit. Instead of reading it, it was enough to just look at it and not translate it into words.

Marc: That's great! Did anyone else find that?

Betsy: Very much the case.

Sarah: Yes. But it wasn't a-syntactic.

Jane: It took me a little while to realize that you wanted them in a specific order.

Marc: So it was a combination of seeing the whole but also reading the syntactic pattern?

Betsy: The context was that I was looking for something that had a set of elements, and I had a film in my mind, so I knew that the book was on top of the table and not that the table was on top of the book. I don't know if that would be the case if I were searching for something.

Marc: The system assumes a certain syntax.

Erin: Sometimes I couldn't figure out the action that was required to get the icons in the right syntax. like, OK, there are stairs, and the guy's walking; YOU figure it out. I couldn't figure out how to say that he was walking "up" the stairs.

[Joshua, Jane, Sandra agree]

Sandra: I found the direction stuff probably the most difficult to deal with, particularly the confusion between direction and screen position. Was I using the right icon for the direction people were moving? I wasn't sure.

Erin: There are lots of other problems. Let me give you an example. There's a guy walking on the sidewalk, and eventually you realize that maybe there's a second where he's walking on the sidewalk, then what comes into view are steps, and then you realize that the steps are connected to the institutional building, and eventually you realize that the institutional building is the Atomic Energy Commission. Now that takes place in maybe three seconds. What do I do? Do I go back and label the institutional building with "AEC"?

Marc: Those are four discrete events that are all happening separately, and of course your knowledge at one is different than your knowledge at another. Having that part of the movie where I don't know that the institutional building is the atomic energy commission would help repurposing.

Jane: There's a reverse angle that I don't know how to indicate.

Erin: I understand your need and your piecing ideas; however I just want to say that then, as a logger, I'm in a position of doing an incredible specificity on almost microsecond by microsecond, and trying to figure out the beginning and end of what events are appropriate descriptions. Like for example, all of a sudden he takes his glasses off; well, prior to that I hadn't thought it was important to say he had glasses on! And so that means I have to go back and say, OK, glasses on, hat on, institutional coat on... luckily for me he didn't mess around with any of his test-tubes, because I don't know where I would have found them.

Jane: I wanted to indicate those big lions on the steps. It looks like a public library and you could easily use that shot as a public library. But I couldn't find any kind of sculpture.

Raphael: I found it, along with scarecrow.

...

Joshua: I thought the stream paradigm was really nice. I liked having something continue until it didn't exist anymore. It felt like a medium-hard math problem that I knew I could solve. There was this real pleasure in taking something and making order out of it.

Jane: When it worked.

Joshua: Yeah but most of the time I managed to say a bare-level thing that you could say, "this is going on, that is going on, this is going on." It's enough that if you needed to go look at it you could see what else is there.

Marc: Do you think others would be able to use that log?

Joshua: Mm-Hmm.



Appendix G

**Media Streams System Designers
and Title Plate**

Media Streams System Designers and Title Plate



Figure 119. The Media Streams System Designers
(left to right: Golan Levin, Marc Davis, Brian Williams)

