UNIVERSITY OF CALIFORNIA, SAN DIEGO

Navigation of Time-Coded Data

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Cognitive Science

by

Adam S. Fouse

Committee in charge:

Professor James D. Hollan, Chair
Professor Maneesh Agrawala
Professor Charles Goodwin
Professor William Griswold
Professor Edwin Hutchins
Professor David Kirsh

2013

The dissertation of Adam S. Fouse is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2013

DEDICATION

To Oliver.

TABLE OF CONTENTS

Signature Page . . . iii

Dedication . . . iv

Table of Contents . . . v

List of Figures . . . viii

Acknowledgements . . . xi

Vita . . . xiii

Abstract of the Dissertation . . . xiv

Chapter 1 Introduction . . . 1

Chapter 2 Observational Research . . . 5
2.1 Types of analysis . . . 6
2.2 Tools to support video analysis . . . 7

Chapter 3 Related Work . . . 11
3.1 Overview . . . 11
3.2 Interactive visualization . . . 12
3.2.1 Visualization of Time . . . 14
3.2.2 Visualization of Time Series . . . 15
3.2.3 Visualization of Video . . . 16
3.3 Event perception . . . 20

Chapter 4 ChronoViz . . . 23
4.1 Design . . . 24
4.1.1 Centrality of time . . . 25
4.1.2 Atomicity of temporal data . . . 25
4.1.3 Flexible visualization . . . 26
4.1.4 Internal and visual representations . . . 28
4.1.5 Implementation . . . 29
4.2 Visualization for Navigation . . . 30
4.3 Categorization and Filtering . . . 33
4.4 Recording annotations . . . 34
4.5 Connecting with Other Tools . . . 35
4.6 Digital Pen and Interactive Paper . . . 37
4.7 Temporal Data Alignment . . . 40
4.7.1 Challenges of Data Alignment . . . 40

4.7.2 Alignment Techniques 42
4.8 Extension 44
4.8.1 Analysis Plugins 45
4.8.2 Visualization Plugins 46

Chapter 5 Data-Driven Design 49
5.1 Participant Observation 49
5.1.1 Research Groups 50
5.1.2 Informed Design 52
5.2 Data Collection 57
5.2.1 Sources of data 58
5.2.2 Supplementary data collection 59
5.3 Data Analysis 60
5.4 Data Overview 64

Chapter 6 Supporting Navigation 68
6.1 Intra-Segment Navigation 69
6.1.1 Scrubbing 70
6.1.2 Re-watching 75
6.1.3 Stepwise Watching 78
6.1.4 Remaining challenges 81
6.2 Creating Annotations 81
6.3 Inter-Segment Navigation 85
6.3.1 Long time-scale scrubbing 86
6.3.2 Identification of Segments 86
6.3.3 Returning to segments 92
6.3.4 Remaining challenges 96

Chapter 7 Visualization of Time-Coded Data 97
7.1 Visualization of Spatiotemporal Data 97
7.2 Leveraging Activity Histories 100
7.3 Visualization Across Multiple Devices 103

Chapter 8 Conclusion 105
8.1 Summary of Thesis Contributions 105
8.2 Future Directions 106

Appendix A Plugin Design 109
A.1 Overview 109
A.2 Analysis Plugins 109
A.2.1 Example Analysis Plugin 111
A.3 Visualization and Data Plugins 114
A.3.1 Example Plugin Code 114
A.3.2 Example View Code 117

A.3.3 Example Javascript Code 122

References 125

LIST OF FIGURES

Figure 2.1: Main display of DIVA video annotation system. Image from (Mackay & Beaudouin-Lafon, 1998). 9

Figure 3.1: Two focus+context visualizations, showing (a) organization structure with a hyperbolic tree browser (from Lamping et al, 1995), and (b) tabular baseball statistics data with the table lens technique (from Rao and Card, 1994). 13

Figure 3.2: Alternate representations of time series to maximize spatial efficiency. Left (image from Heer et al, 2009): (a) standard filled time-series graph, (b) mirrored time-series graph, (c) horizon graph. Right (adapted from Bade at al, 2004): Multiple levels of semantic zoom for a single underlying time-series. 16

Figure 3.3: Video timeline created from a column of pixels from each video frame (Nunes et al, 2006) 19

Figure 3.4: Using an arrow, as in a video storyboard, to visualize motion in video (Goldman et al, 2006) 19

Figure 4.1: Example ChronoViz session showing flight data collected from Ed Hutchins' research, including multiple videos, simulator data, transcript data, eye-tracking data, and notes taken with a digital pen. . . 31

Figure 4.2: Spatiotemporal visualization in ChronoViz: (a) Gaze data overlaid on video recorded from eye-tracking glasses, (b) Motion tracking data of two dancers recorded with a Microsoft Kinect, and (c) Elephant movement trajectories entered with a digital pen (image enhanced for print legibility). 32

Figure 4.3: Interface for editing timeline filters. 34

Figure 4.4: Entering annotations in ChronoViz with the (a) Annotation Inspector or (b) Annotation Quick Entry pop-up window. 36

Figure 4.5: Digital notes in ChronoViz shown as: (a) an individual note attached to an annotation and (b) a page of notes. 38

Figure 4.6: Video playback controls printed on Anoto paper. A user can tap on the controls with a Bluetooth Anoto pen to control video playback in ChronoViz. 39

Figure 4.7: Alignment of time-series data by clicking on a point on the visualized time-series line graph and selecting "Align to Playhead". . . . 43

Figure 4.8: Alignment of digital notes by selecting a mark made with the digital pen. In this case, a mark was made in a specially designed "alignment" region. 44

Figure 4.9: Discursis visualization of conversation analysis (right), shown within a ChronoViz window and linked to data in an existing ChronoViz session (left). 48

Figure 5.1: Three levels of inspection with notes. The notes are represented as marks on a page, annotations on a timeline, and an image in the pop-up window. 53
Figure 5.2: Comparison of line graph (top) and color-mapped (bottom) visualizations for binary time-series data. 56
Figure 5.3: A frame from a video of Professor Ed Hutchins using ChronoViz for analysis, showing arrangement of windows on screen and a pointing gesture in relation to the displayed data. 60
Figure 5.4: Software tool created to support analysis of ChronoViz interaction logs. On the left is a hierarchical list of logs, arranged by user then by video. In the middle is a plot of a selected log file. On the right is a summary of various measures that are computed from the log file. 61
Figure 5.5: An example video activity graph, showing a researcher watching a 14-second segment of video twice over the span of about two minutes of analysis activity. 63
Figure 5.6: Distribution of the percentage of a video viewed analysis sessions. . 65
Figure 5.7: Distribution of the duration of navigation activity during analysis sessions. 53 logs with more than 60 minutes of activity are not shown. 66
Figure 5.8: Number of sessions for each video viewed by each researcher. . . . 67

Figure 6.1: Histogram of direction changes during scrubbing. 73
Figure 6.2: Video activity graph, showing re-watching activity for two segments of video. 77
Figure 6.3: Video activity graph, showing stepwise watching activity leading to the creation of two annotations, as represented by red dots. 79
Figure 6.4: Video activity graph, showing stepwise watching activity with intermediate short scrubs. 79
Figure 6.5: Early version of ChronoViz showing an automatically generated timeline with frames from a selected annotation. 82
Figure 6.6: Exporting a video clip by selecting an existing annotation. The submenu gives the option of exporting a clip from one of three videos that are associated with the data set. 83
Figure 6.7: Visualization of four binary time series, showing data from four regions of gaze data from a flight simulation. Bright colors represent gaze directed at that region, dark grey represents gaze directed away from that region, and transparent represent missing data. 88
Figure 6.8: Color-mapped time-series visualization, adapted to show presence of image data to aid navigation of data recorded with a SenseCam. . 88
Figure 6.9: Transcript of two pilots, with utterances of each pilot shown as a row of annotations on a ChronoViz timeline. 90
Figure 6.10: Audio angle data recorded with a Microsoft Kinect of a conversation between two people, pre-processed to create annotations to represent the conversation. 91

Figure 6.11: Results of running a plugin to compute a measure of average writing speed (shown as a line graph) from digital notes data. 92
Figure 6.12: Searching in a transcript from a simulated flight for occurrences of the word "flaps". 93

Figure 7.1: Selection of a spatial region of notes generates a timeline below the notes, showing the time points when those notes were made. 99
Figure 7.2: Selection of a spatial region of eye tracking data, with transition percentages shown for a selected region. 100
Figure 7.3: Video player with timeline showing activity-based visualization. Frames from segments of the video that have been viewed more are visible on the timeline, while unseen segments are occluded. . . 101
Figure 7.4: Top: Design for visualizing temporally-linked activity oriented around related segments. Bottom: Visualization design for showing a segment overview from temporal boundaries. 102

Figure A.1: Plugin configuration window that allows the user to specify a data set and parameters before running the plugin. 111

ACKNOWLEDGEMENTS

As with any large research project, this thesis would not have been possible without the collaboration and support of many others. In particular, two people had substantial impact on this thesis, and without whom little of the work would have been possible. First and foremost is my advisor, Jim Hollan. Throughout my time as a graduate student, Jim has encouraged me to follow my intuitions and gently nudged me to keep me on course. His vision and attitude have inspired me to not only do good research but also to enjoy the doing it. Only in hindsight do I realize how accurate his early guidance was. Second is Ed Hutchins. In many ways this thesis can trace its origins to a day in the summer of 2009 when Ed asked if my framework for running video segmenting experiments could be adapted to enable text annotations of the videos. My interaction with Ed has been greatly rewarding, and our continuing collaboration enabled me to fully explore the possibilities provided by advanced visualization to aid analysis. Ed's early belief in the potential of ChronoViz gave me the needed confidence to continue with the development of advanced ways to interact with data. I would also like that thank the other members of my committee. Thank you to Maneesh Agrawala, Chuck Goodwin, Bill Griswold, and David Kirsh for your feedback and suggestions.

There are many people in the UCSD community that have provided encouragement and feedback for my work, especially the members of the Distributed Cognition and Human-Computer Interaction Laboratory. I believe that this thesis would not have been possible in any other environment. In particular, I would like to thank Nadir Weibel, my collaborator on all of the digital pen and interactive paper technology. In addition to being a fantastic collaborator, Nadir provided valuable feedback on ChronoViz and excellent research advice. Thanks also to Whitney Friedman, who was involved with ChronoViz from the very beginning and provided consistently great feedback. Whitney used ChronoViz for analysis in both the aviation and non-human cognition domains and helped me to see its broad potential. Other members of the lab provided helpful guidance and feedback, especially my fellow graduate students past and present: Gaston Cangiano, Anne Marie Piper, and Nan Renner.

This work would not have been possible without the people that actively used ChronoViz. I interacted with the aviation research group on an almost daily basis, and its members contributed much insight and many ideas. In addition to those already thanked, Colleen Emmenegger, Sarah Kimmich, and Lara Cheng deserve mention. Thanks to the elephant research team, especially Chris Johnson and Kelly Inglett, for your excitement and patience for new technology.

Last, and perhaps most importantly, I would like to thank my family for their love and support. My parents Scott and Melody have always encouraged me to pursue my dreams. My brother Shaun and sister Shannon have done their own amazing research and continue to provide a measuring stick for my own success. Finally, I want to thank my wife Andrea and my son Oliver. Thank you, Andrea, for your love and and for supporting me when graduate school was frustrating and challenging. Thank you, Oliver, for inspiring me to finish and providing smiles that make the worst days suddenly better.

VITA

2004	B. A. in Cognitive Science and Computer Science, Brown University
2010	M. S. in Cognitive Science, University of California, San Diego
2013	Ph. D. in Cognitive Science, University of California, San Diego

PUBLICATIONS

Adam Fouse, Nadir Weibel, Christine Johnson, and James D. Hollan. "Reifying Social Movement Trajectories." Proceedings of ACM Conference on Human Factors in Computing Systems (CHI) 2013, *to appear*.

Nadir Weibel, Adam Fouse, Colleen Emmenegger, Whitney Friedman, Edwin Hutchins and James D. Hollan. "Digital Pen and Paper Practices in Observational Research." Proceedings of ACM Conference on Human Factors in Computing Systems (CHI) 2012.

Nadir Weibel, Adam Fouse, Colleen Emmenegger, Sara Kimmich and Edwin Hutchins. "Let's look at the Cockpit: Exploring Mobile Eye-Tracking for Observational Research on the Flight Deck." Proceedings of ACM Symposium on Eye Tracking Research and Applications (ETRA) 2012.

Adam Fouse and James D. Hollan. "Visualization of Exploratory Video Analysis." Proceedings of IEEE Symposium on Information Visualization, Poster Session, 2011.

Adam Fouse, Nadir Weibel, Edwin Hutchins and James D. Hollan. "ChronoViz: A system for supporting navigation of time-coded data." Extended Abstracts (Interactivity Track) of ACM Conference on Human Factors in Computing Systems (CHI) 2011.

Nadir Weibel, Adam Fouse, Edwin Hutchins and James D. Hollan. "Supporting An Integrated Paper-Digital Workflow for Observational Research." Proceedings of International Conference on Intelligent User Interfaces (IUI) 2011.

Adam Fouse and James D. Hollan. "DataPrism: A Tool for Visualizing Multimodal Data." Proceedings of 7th International Conference on Methods and Techniques in Behavioral Research (Measuring Behavior) 2010.

Joshua M. Lewis, Adam S. Fouse, Virginia R. de Sa. "Cross-Modal Influence on Binocular Rivalry." Proceedings of the 32nd Annual Conference of the Cognitive Science Society, 2010.

Bisantz, A., Stone, R., Pfautz, J., Fouse, A., Farry, M., Roth, E., Nagy, A., and Daniels, G. "Visual Representation of Meta-Information." Journal of Cognitive Engineering and Decision Making, 2009, 3(1), pp. 67-91.

ABSTRACT OF THE DISSERTATION

Navigation of Time-Coded Data

by

Adam S. Fouse

Doctor of Philosophy in Cognitive Science

University of California, San Diego, 2013

Professor James D. Hollan, Chair

Advances in technology now make it possible to capture detailed multimodal data about real-world everyday activity. Researchers have taken advantage of these advances to address questions about activity in more systematic and precise ways. Along with exciting opportunities to record data in ways that were not possible before, there are also analysis challenges due to the quantity and richness of the data. New tools and techniques are needed to address the challenges of analyzing dynamic multimodal activity data. I address the question of how such tools should be designed by developing an understanding of the data navigation patterns exhibited by researchers while they perform analysis.

This thesis presents ChronoViz, a novel tool to support navigation, visualization, annotation, and analysis of multiple streams of time-coded data. ChronoViz supports

analysis of data through the use of synchronized interactive visual representations of multiple data streams. Since visualizations are linked by time, each data stream can be used for navigation of the data set as a whole, and visualizations can flexibly be added, configured, and arranged to suit individual analysis needs.

ChronoViz served three functions in this research. It enabled collaboration with researchers who collect observational time-based data, recorded the researchers' activity during use of ChronoViz for their data analysis, and provided a platform to enable design of interactive visualizations. Researchers used ChronoViz in their existing research projects, so the activity recorded was that of real analysis efforts. I describe patterns of navigation, how specific designs support those patterns, and identify ways to better support them in the future.

Chapter 1

Introduction

> *The direct goal is to see patterns in the data and understand the overall behavior, but the more accurately the data are visually decoded, the better our chance to detect and properly understand the patterns and behaviour of the data.*
>
> (Cleveland & McGill, 1987)

Human interaction involves a complex coordination of multimodal information. We combine information from multiple senses and use multiple sources of information to understand things in the world around us. The way the people interact with recorded information has historically been less rich, often taking the form of text and pictures. As such, the approach that scientists have taken to understand human perception and processing of multimodal recorded information has often taken the form of models, such as cognitive load theory, that consider information processing in terms of the capacities of sensory pathways of the brain. These have been useful for generating some insight into the way that people process and integrate multiple sources of information.

However, as technology moves forward, the type of information that can be recorded for later interpretation includes more dynamic and complex information. In addition to static text and images, we may now deal with dynamic multimodal information, potentially incorporating multiple sources of audio and video, large sets of numeric data, sensed data from our physical interactions, and complex interactive displays of such information. These sources of data represent not only a movement toward dynamic forms of information, but also a vast increase in quantity. Due to these changes, it may be more productive to study human interaction with such information in a similar way to recent research on multimodal human interaction with others: by considering it as a culturally situated, distributed cognitive system (Suchman, 1987; Hutchins, 1995).

One group of people that especially experience these changes in recorded information is the community of researchers that study real-world activity. Advances in technology have made it possible to easily record more data about activity in ways that were not possible before. Consider the following three examples: Affordable high-definition video recording and large inexpensive digital storage have encouraged recording of detailed video from multiple angles, enabling scrutiny of small details of activity. Small wearable sensors have made it possible to get detailed records of human movement, enabling signal processing techniques that can identify trends and anomalies in activity patterns. Interaction with digital systems can produce detailed logs of interaction or performance, enabling both detailed and widespread collection of records of digitally-mediated activity. Each of these types of data presents the possibility to answer questions about activity in more systematic and precise ways. The ability to collect these rich sources of data represents an unprecedented opportunity for scientific scrutiny of activity as it occurs in real-world, everyday situations.

Along with the possibilities created by new data collection abilities, there is also a clear challenge of analysis. Some forms of data, such as video, can be tedious and time-consuming to analyze. Other forms of data, such as sensor readings, may only be useful in combination with other data about the same activity. While each type of data of its own may be valuable, the most valuable insights may be created by combining multiple multimodal data recordings of the same activity. New tools are needed to address the challenge of analyzing and synthesizing the information across data sources and across time, and an understanding of how these researchers interact with these complex sources of information is needed to develop such tools.

This thesis seeks to answer the question of how such tools should be designed. While there are many possible ways that one might go about answering this question, the research described in this thesis is based primarily on understanding the activity of researchers as they perform analysis. This consists of observing, recording, and analyzing data about the real-world use of multimodal information visualization for the purpose of scientific analysis. Basing design on this understanding of analysis activity addresses some limitations of other methods. Tightly controlled experimental methods lend themselves well to inferential statistics and looking at performance of many people across identical settings. However, the types of data that are currently being considered have a combinatorial complexity due to the combination of multiple different types of visualization at the same time; experimental designs would rapidly increase in complexity. More importantly, collecting data from real-world usage circumvents the need to provide artificial motivation. The activity I observed and the data I collected were the result of activity that was based on existing motivations for the researchers' analysis efforts.

The theoretical framework of distributed cognition (Hutchins, 1995; Hollan, Hutchins, & Kirsh, 2000) provides an ideal perspective for understanding analysis activity as it occurs in real-world research efforts. Distributed cognition expands the unit of analysis beyond the individual to consider interactions within a system of activity, potentially involving multiple people, artifacts, and cultural practices. In considering researchers interacting with complex data though interactive systems for the purpose of scientific research, we need to consider how each element of the system affects the visible activity, and how the combination of a researcher (or researchers) interacting with data as mediated through the system produces the result.

Although the data I collected and analyzed was from actual analysis efforts, these were analysis efforts that were directly influenced by me as I produced continually evolving technology to aid these efforts. The researchers used a software tool that I designed and developed called "ChronoViz". This tool is used for visualization and analysis of the types of data that are being considered, and is described in detail in Chapter 4. Two primary considerations drove the development of this tool. The first consideration was to create an environment to support exploration and evaluation of visualization of heterogeneous sets of time-coded data. This is accomplished by the extensible visualization capabilities of ChronoViz and by the ability to collect detailed logs of interaction. The second consideration was to enable working with these types of data in a more effective way than was provided by contemporary tools.

To understand the motivation behind developing a new tool rather than relying on an existing tool for visualizing this type of data, it is necessary to first discuss the ecology of tools and common practices for annotation of video that existed at the time this work was begun. The next chapter of this thesis describes both the practices and tools for video annotation. Following this discussion of this domain, related work from computer science, information visualization, and experimental psychology is discussed in Chapter 3. The remainder of the this thesis describes my research and results. Chapter 4 provides a description of the design and capabilities of ChronoViz. Chapter 5 describes in detail the methods for data collection. Chapter 6 describes the temporal navigation patterns of researchers using ChronoViz, and how those patterns are supported through the design of ChronoViz. Chapter 7 describes additional advances in visualization to support data navigation. Finally, Chapter 8 summarizes the contributions of this research and offers guidance for future work in this area.

Chapter 2

Observational Research

While scientific research is moving toward complex sets of multiple different forms of data, it is worth first considering current practice. One characteristic of current practice, especially for the type of observational research that is the domain of scrutiny of this thesis, is the primacy of video. Digital media (i.e., video with or without sound) is the primary source of data for much of observational research in the social sciences.

Video provides a rich, veridical account of activity, and supports a wide range of analytical approaches, from identifying patterns of behavior to looking at frame-by-frame temporal dynamics. Video recording is also increasingly inconspicuous, inexpensive, and ubiquitous, as a result of rapidly advancing video recording technology. While researchers have made use of video for as long as video recording technology has been around, the improvements in video recording technology have increasingly allowed researchers to use video analysis as a primary method for identifying and quantifying phenomena. Because of this, the process of video analysis and tools to support this process has been widely studied.

One step in the these efforts has been to identify the elements of a video research workflow, and how each of these activities should be supported by software tools. Pea and Hoffert (2007) offered an end-to-end description of a workflow for using digital video as a source of data for research. This workflow includes the stages from initial planning to final publication, and gives a detailed focus to an iterative process of analysis. Pea and Lemke (2007) made further mention of the iterative nature of video research, noting the need to "chunk" the video records into segments that may be defined in a number of ways.

2.1 Types of analysis

Analysis of observational data has a rich history spanning several traditions. Sanderson and colleagues (Sanderson & Fisher, 1994; Sanderson et al., 1994) coined the term Exploratory Sequential Data Analysis (ESDA) to refer to examination of activity as it unfolds over time. This includes analysis of any type of log or recording of "raw sequences" of activity, which depending on the setting may include things like computer logs and video. To help understand ESDA and how technology can affect analysis efforts, they discuss "Sequence Time:Activity Time" (ST:AT) ratios. These are ratios of the duration of raw sequence time recorded to the duration of time spent on analysis. High ST:AT ratios are often an impediment to productive ESDA, but Sanderson and Fisher note that software for supporting ESDA can both "exacerbate and alleviate this problem" (Sanderson & Fisher, 1994).

The video analysis activity that I consider largely falls into the what Sanderson and Fisher call the "Social Tradition", where the goal is to "understand the social, organizational, and material worlds within which individuals and groups operate" (Sanderson & Fisher, 1994). Within this type of practice, two broad categories of activity can be defined to be *exploratory analysis* and *coding*. These two types of activity involve different stated goals. With exploratory analysis, researchers perhaps have some ideas of the nature of the recorded data, and the nature of the general class of activity under scrutiny, but no specific coding scheme that guides their activity. In these situations, it is often necessary to view data multiple times, at different speeds, and from multiple perspectives (Sanderson & Fisher, 1994). Exploratory analysis often involves more scrutiny to a smaller corpus of data.

On the other hand, coding involves recognition of a set of specific pre-defined activities. These pre-defined activities may have been determined by an earlier exploratory analysis phase, but they may also have be defined beforehand as part of a study design, or based on earlier studies. Often, these coding schemes are evolved over the course of a research project (Weston et al., 2001).

For a given video analysis project, exploratory analysis and coding can often be iterative. Hypotheses about the data may be formed from exploratory analysis, and these hypotheses may be tested through coding of a larger corpus of data. As a coding scheme is applied, problems with the scheme may be found that necessitate revisiting data in a more exploratory fashion. Upon completion of coding a large corpus of data, analysis of the coded data may lead to new questions that aren't answerable with the existing code schema, and require more exploratory analysis to develop and refine new coding schemes.

A hypothesis is that these two types of activity not only involve different goals, but quantitatively different patterns of activity. While there are many low-level tasks that are common to each type of activity, such as the need to control video playback and record observations about what is seen in the video, the details of how these tasks are performed, why they are performed, and the order in which they are performed are likely to be different. As noted above, there are often smooth transitions between different styles. Classifying analysis sessions as exploratory analysis or coding can be problematic, so identifying activity patterns as belonging to one category or another is not productive, but rather a goal should be to design tools to support a continuum of analysis styles and transitions between then. For example, when entering annotations, researchers performing exploratory analysis may want a high degree of flexibility for recording and linking observations, while researchers performing coding may want more structure to make it easy to quickly choose among the established codes. Enabling a range of annotation entry possibilities could provide better support for the entire course of an analysis effort.

2.2 Tools to support video analysis

While a range of activities are involved in observational data analysis, the task of video annotation has received particular focus, both from researchers and software developers.

The requirements for the task of video annotation have also been studied to be able to better design and compare software tools. Hagedorn, Hailpern, and Karahalios (2008) drew on interviews with experienced video researchers to identify a set of requirements for effective video annotation software based on current practices. Among these requirements was the need to facilitate coding workflow, capturing and displaying appropriate types of additional data, and flexible forms of video playback. Identification of these requirements was driven by the development of the *VCode* and *VData* pair of video analysis tools.

Hofmann, Hollender, and Fellner (2009) also established a set of capabilities needed to support the video research process. The general categories of tasks they defined are *configuration*, *segmentation*, *annotation*, *exploration*, and *externalization*. Among the specific requirements they identified, the ones particularly germane to the current discussion are supporting the annotation style of individual users, synchronization of data sources, and supporting exploration by providing multiple ways of navigating the data. This set of requirements was designed to be used as a comparison across a range of video annotations tools, including *ELAN* (Wittenburg et al., 2006), *The Observer* (Zimmerman et al., 2009), *Transana* (Woods & Fassnacht, 2007), and *WebDIVER* (Pea et al., 2004), among others.

As advances in technology have affected data collection efforts, similar technological advances have enabled improved tools to support analysis. Many early tools that provided computational support for observational data analysis focused their efforts on entry and analysis of codes. These typically provided facilities to enter structured, time-coded annotations or codes, and to support statistical analyses of those codes, with support for having this activity linked to video. Examples of these tools include MacSHAPA (Sanderson et al., 1994) and VideoNoter (Roschelle & Goldman, 1991).

Further efforts to support observational data analysis took more advantage of the increased availability of digital multimedia and dynamic graphical displays. For example, the DIVA project (Mackay & Beaudouin-Lafon, 1998) used a three-dimensional representation of streams of annotations. In comparison to many earlier efforts, DIVA represented annotations as “streams” rather than “chunks” of information, to better match the continuous nature of video. These streams of annotations were translated into streams of color that moved along the edges of a video, as shown in Figure 2.1. While earlier systems required video to be played on an external device, DIVA was able to integrate the display of video and annotations.

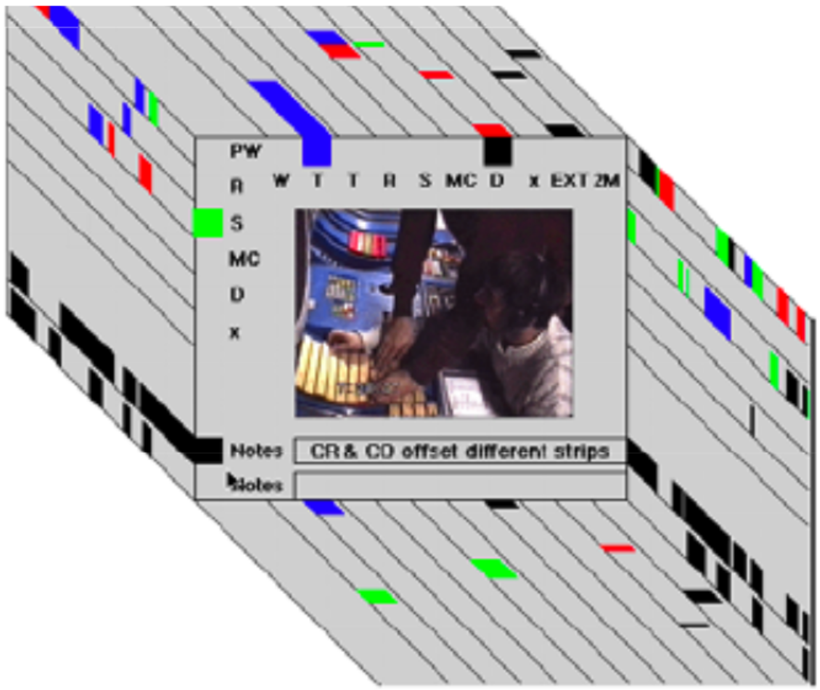


Figure 2.1: Main display of DIVA video annotation system. Image from (Mackay & Beaudouin-Lafon, 1998).

In response to the growing use of video as a research tool, a number of modern tools are available to support research tasks involving video. A central task that many of these tools support is systematic annotation of video. Basic video players, such as QuickTime Player or Windows Media Player, are often used in combination with a speadsheet program (e.g., Microsoft Excel) to record times and annotations. This has the advantage of using individual tools that may be quite familiar, and supporting a very flexible workflow. However, this workflow can also be very tedious. Even the act of recording times by manually copying the timecodes displayed in the video player into a spreadsheet can unnecessarily burden the annotation process, as well as potentially introduce error. To address these concerns, as well as to better support creation of annotations that comply with structured coding schemes, a number of tools have been developed to explicitly support video annotation. These include tools that are specialized for transcription of audio, such as InqScribe[1], Transcriva[2], and ExpressScribe[3]; tools to support collaborative video analysis such as Diver (Pea et al., 2004); tools that support creation of annotations by structured categories and with limited support for additional non-video data sources such as ELAN (Wittenburg et al., 2006), Transana (Woods & Fassnacht, 2007), and VCode (Hagedorn et al., 2008); and tools that offer extensive support for multiple data sources such as Noldus Observer (Zimmerman et al., 2009), Mangold Interact[4], and Anvil (Kipp, 2008).

[1]http://www.inqscribe.com
[2]http://www.bartastechnologies.com/products/transcriva/
[3]http://www.nch.com.au/scribe/index.html
[4]http://www.mangold-international.com/

Chapter 3

Related Work

3.1 Overview

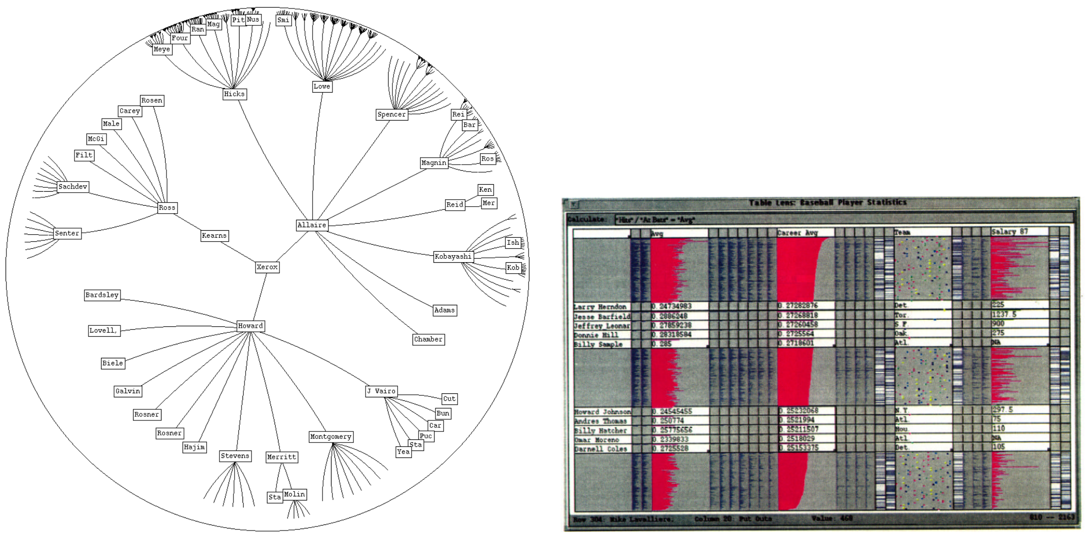
Supporting navigation of temporal data through the use of interactive visualization benefits from understanding of both related work on interactive visualization and related work on perception and understanding of video. Interactive information visualization as a research field can trace its roots to the introduction of graphical computer systems. Within this field, this is some work that is directly relevant, looking at visualization of time series and visualization of video, that can provide important context for the work in this thesis. For work on understanding video, we can turn to work on event perception. This field focuses on how humans perceive events, usually concerned with visual perception of events. Since perception of events in video is one of the basic elements of video analysis, this research provides an important starting point for understanding the activity of video analysis. These two fields offer complementary grounding for this problem, with event perception providing a cognitive perspective on basic elements of video interaction, and information visualization providing existing techniques that have been used to address similar problems.

3.2 Interactive visualization

While existing tools support working with video or even multimodal streams of data in a variety of ways (as discussed in Section 2.2), one of the areas where these tools are generally lacking is providing powerful interactive visualization facility. Tukey identifies visualization as a critical component of exploratory data analysis (Tukey, 1977), and several researchers have noted the application of Tukey's ideas to analysis of sequential data such as video (Sanderson & Fisher, 1994; Mackay & Beaudouin-Lafon, 1998). In this section, I discuss visualization research that has broad application, then discuss directly relevant work that focuses on timeline visualization, time-series visualization, and video visualization.

With the advent of graphical digital systems, researchers quickly recognized the potential for providing dynamic visual representations of information, and the potential for these representations to shape our understanding of that information. Many researchers have identified some of the cognitive benefits of of visualization (Card, Mackinlay, & Shneiderman, 1999) as well as the benefits of basing the design of visualization on knowledge of the strengths and limitations of the human visual processing system (Ware, 2004).

One of the strengths of providing *interactive* visualization facilities, as opposed to designing for a static medium such as print, is that there is no need for a view of data to be a "one size fits all" solution. Views can be changed based on the particular needs of the user, and multiple views can be provided to show different perspectives of the data simultaneously. One aspect of this approach that has received much attention is the *focus-context* problem, which refers to the problem of how to present detailed information while retaining awareness of the larger context of information. Some techniques, such as fisheye views (Furnas, 1986), hyperbolic tree visualization (Lamping, Rao, & Pirolli, 1995), and the table lens (Rao & Card, 1994), use a single view that contains both focus and context, and transforms the visual display as the focus is shifted. These techniques show the area of focus in high detail, while showing surrounding information in progressively less detail in proportion to some distance from the item of focus. Figure 3.1 shows these techniques as applied to network and tabular data. Other techniques, such as the multiscale zooming found in Pad++ (Bederson & Hollan, 1994), aim



(a) (b)

Figure 3.1: Two focus+context visualizations, showing (a) organization structure with a hyperbolic tree browser (from Lamping et al, 1995), and (b) tabular baseball statistics data with the table lens technique (from Rao and Card, 1994).

to address the problem by providing smooth, fast transitions between different levels of detail. Zooming approaches rely on the user's natural ability to integrate information when moving through a space, as when moving through the physical world. While zooming offers advantages for integrating data a multiple scales, separate views at different scales can enable easier comparisons (Plumlee & Ware, 2006). Multiple window approaches, such as the Improvise system (Weaver, 2004), Snap-Together visualizations (North & Shneiderman, 2000) and linked 3D views (Plumlee & Ware, 2003), rely on multiple different views to provide both focus and context without distortion. These are specific instances of a general approach to visualization referred to as "multiple coordinated views" (Wang Baldonado, Woodruff, & Kuchinsky, 2000), using multiple views to support exploration of relationships between different parts of a data set or different types of data.

In addition to the problem of how focus and context can be visually presented, it is important to consider how interactive information visualization techniques can assist exploration of the data. A number of techniques have been identified and evaluated with this aim. "Brushing", a technique where data that is manually selected (either by visual dragging in one view or selection by parameters) is simultaneously highlighted in all views, can be effective in working with multiple scatterplots (Becker & Cleveland, 1987) as well as different types of graphs, as used in the GGobi visualization system (Swayne, Buja, & Lang, 2003). Roberts (2007) provided an overview of additional techniques to explicitly support data exploration with multiple coordinated views, including the use of alternate representations, dynamic queries (Shneiderman, 1994), and brushing.

There is growing acknowledgement of the importance of evaluation of information visualization methods (Plaisant, 2004), and especially of perception-based evaluations (House, Bair, & Ware, 2006). These methods are often based in the application of known perceptual properties to the design of visualization. One characteristic that has often been evaluated is the accuracy with which information is transferred. For example, a color scale that is linear in numeric value may not be perceived as linear in the information it represents (Ware, 1988; Silva, Madeira, & Santos, 2007). Further, the chosen scale for representing information may be too fine grained to provide benefit for information perception, and may simply add clutter without improving perception of information (Bisantz, Marsiglio, & Munch, 2005).

3.2.1 Visualization of Time

Multiple efforts have attempted to identify the space of time-oriented visualizations (Aigner, Miksch, Schumann, & Tominski, 2011; Muller & Schumann, 2003). While it can be difficult to directly compare time-oriented visualizations that are application specific to those that are more general, there three components that can be considered: characteristics related to data, time, and visual mappings. Data can be abstract or spatial, and univariate or multivariate. Time can be conceptualized and visualized as linear or cyclic, and as points or intervals. Visual mappings can be static or dynamic, and two-dimensional or three-dimensional. For example, a time-series line graph is typically used to visualize abstract univariate data, with linearly arranged time points.

Much work on time-based visualization has focused on representations of personal events on timelines. Many of these efforts have focused on showing patterns in a single person's events, such as for reviewing a medical history (Plaisant, Milash, Rose, Widoff, & Shneiderman, 1996). Others have focused on revealing relationships between multiple records, such as comparing across a number of personal histories to find common patterns (T. D. Wang et al., 2008). One technique that has been used by several projects is to use visual abstractions to summarize data across multiple records, such as with histograms (André et al., 2007) or tree structures (Wongsuphasawat et al., 2011).

Among the challenges faced by designers of interactive time-oriented visualizations are supporting exploration at different scales and revealing relationships between different parts of the data. Representing and interacting at different time scales is a common problem among many different types of time-coded data, and a specific instance of the general visualization problem of focus+context described in the previous section. Users of time-oriented visualization commonly need to have large scale navigation capabilities to move among time segments as well as small scale navigation capabilities to have precise movement within segments. Some approaches to this problem have focused on how users interact with timelines, and include the use of coordinated timelines at multiple scales (Richter, Brotherton, Abowd, & Truong, 1999) and content-aware timelines that decouple video playback rate from mouse movements (Pongnumkul, Wang, Ramos, & Cohen, 2010).

3.2.2 Visualization of Time Series

A substantial portion of information visualization research concerns the perception and interpretation of graphs. One particular type of graph that is relevant to the proposed research is the time-series graph, where time is represented on the horizontal axis, a relevant variable is represented on the vertical axis, and data points consisting of values for a given time are connected with a line.

Before considering time-series specifically, it is worth considering general background on graphs. One element of graph perception that is particularly relevant to the perception of time-series graphs is the perception of rates of change, since time-series analysis is often directly concerned with patterns of change over time. Cleveland, McGill, and McGill (1988) demonstrated that differences in line slope can best be detected when the average slope of the lines is 45 degrees. More recently, Heer and Agrawala (2006) demonstrated computational techniques to automatically optimize graphs for the “45-degree rule.”

Looking specifically at time-series, Heer et al (2009) demonstrated the effec-

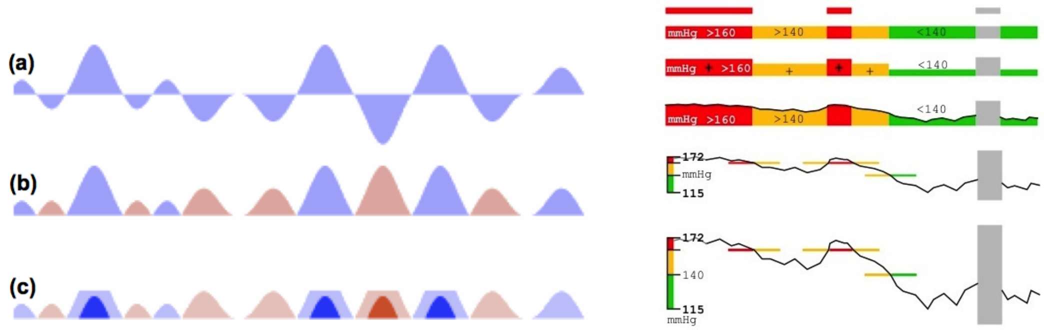


Figure 3.2: Alternate representations of time series to maximize spatial efficiency. Left (image from Heer et al, 2009): (a) standard filled time-series graph, (b) mirrored time-series graph, (c) horizon graph. Right (adapted from Bade at al, 2004): Multiple levels of semantic zoom for a single underlying time-series.

tiveness of *horizon graphs*, where different levels of the vertical axis are layered upon each other, as shown in Figure 3.2. These graphs can be used to provide approximately the same amount of detail as a normal line graph, but in a smaller vertical space. Another approach for reducing the vertical space required to display data is to use semantic zooming techniques and make a qualitative change in the representation for different levels of detail. Bade et al (2004) explored a continuum of representations for medical time-series data, with colored bars smoothly transitioning to detailed line graphs, also shown in Figure 3.2. These different scales of presentation were designed to better support doctors' activity as well as use display space more efficiently. The more compact representations were designed to give a faster overview of the data while allowing the doctors to zoom for more detail if needed.

3.2.3 Visualization of Video

While there are many forms of data to consider in this research, video (or sequences of still images) calls for special attention due to the central nature of video in much of the research that was being considered, and to the difficulty in making sense of a large amount of video data.

A growing number of techniques have been developed to aid in automatic video summarization. Two general categories of summarization techniques are those that generate a summary video and those that generate a summary image. Summary videos summarize a longer video by creating a shorter video that contains important points in the video. To determine which segments of video to include in the summary, these types of techniques use a number of computer vision techniques such as optical flow estimation and motion processing, often in combination with audio and visual recognition techniques for identifying keywords or objects (Smith & Kanade, 1998). Once segments have been identified, a second challenge is deciding how to combine them. One effort used these techniques to identify important areas in the video, and then use constant rate frame dropping to shorten other areas (Li, Zhang, & Tretter, 2001). Another effort combined segments from different times by not only determining relevant times but also relevant regions of the frame (Kang, Chen, Matsushita, & Tang, 2006). A similar automatic summarization effort used spatio-temporal analysis to identify repeating patterns and select important shots, which were then extracted from the video to form a summary (Xiao, Wang, Pan, & Wu, 2008).

Summary images often take the form of digital photomontages (Agarwala et al., 2004). These are composite images with a set of images (in the case of video, frames) that are combined with the goal of creating a seamless single image, usually through intelligent panoramic stitching of images in combination with various image processing techniques. Early efforts for creating summary images from video focused on selection of representative frames (Truong & Venkatesh, 2007) and techniques for combining frames into a single image (T. Wang, Mei, Hua, Liu, & Zhou, 2007). More recent efforts have focused on techniques for interacting with summary images. This includes making photomontages dynamic by offering possibilities for scrubbing over parts of the image to reveal movement (Correa & Ma, 2010), or enabling continuous zoom on photomontages to allow users to see more detail for segments of interest (Barnes, Goldman, Shechtman, & Finkelstein, 2010).

Other summarization efforts take a more semantic approach. The Media Streams project (Davis, 1993, 2000) defined an iconic language to use for manually created icon-based summary timelines. Relying on iconic representations creates a uniformity so that collections of videos can be consistently annotated and compared. Another effort combined computational vision techniques such as shot boundaries and visual clustering with a scene transition graph, which can then be presented to users as "story units" to browse (Yeung, Yeo, & Liu;, 1996). In this case, a video is broken up into shots, and the shots are clustered based on visual similarities and temporal constraints. The user can then find similar clips (e.g., shots of the same person in a given scene), and follow a path through the video from scene to scene. Another similar approach that is specialized for moving overhead video, such as might be taken from an airplane, uses visual alignment between frames and motion analysis to segment a video and select interesting clips based on differences in motion patterns, then construct a spatial mosaic image of the area the video covers, which can then be combined into layers of static mosaic image and moving video and presented to the user (Pope, Kumar, Sawhney, & Wan, 1998). The Vis-A-Vis system (Romero, Vialard, Peponis, Stasko, & Abowd, 2011) takes similar advantage of spatial information for the activity recorded in video and combines it with computer vision techniques to visualize human activity in houses.

Visual representations of video content have also been created. These tend to rely on human abilities to make sense of a condensed version of the entire video, rather than selecting specific points in the video. The Recreating Movement project (Hilpoltsteiner, 2006) demonstrates a technique of stacking frames on top of each other in three-dimensional space. This technique is particularly applicable to the analysis of physical movements. In videos with a single human subject, a chroma-key is used to isolate the subject from the background, and can reveal a range of movement at a single glance much better than with a traditional still frame or set of frames. A more abstract technique for visually representing the content of a video is to use slices of visual information from each frame (Davis, 2000; Nunes, Greenberg, Carpendale, & Gutwin, 2006). In this case, the visualization uses a single column of pixels near the center of each frame that are then combined to create an abstract timeline representation of the video, as shown in Figure 3.3. The contiguity and visual flow of colors provides an indication of areas of activity and change, providing a high-level overview of the data and aiding further analysis.

Additional methods that have been studied for representing video content use stylized representations of frames to create images that are easier to understand. These frequently rely on conventions that have been developed for representing motion and action in still graphical form, such as those used in comics as described by

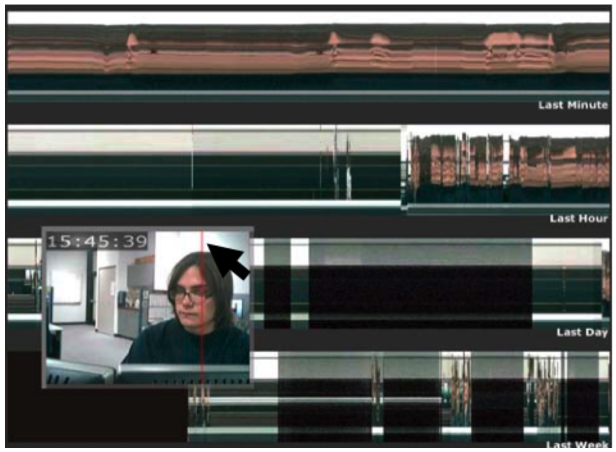


Figure 3.3: Video timeline created from a column of pixels from each video frame (Nunes et al, 2006)

McCloud (1994). In work by Goldman et al (2006), video storyboarding techniques were applied to create schematic storyboards. These are static visual representations that utilize stitched frames and graphic forms such as arrows (shown in Figure 3.4) and lines to create annotated storyboards describing the action in a video. The generation of these schematic storyboards is a largely automated process combining computer vision, rendering, and visualization computational techniques. Partially in response to increased availability of low-cost motion capture devices such as Microsoft Kinect[1], similar efforts have been applied to visualize motion capture data (Bouvier-Zappa, Ostromoukhov, & Poulin, 2007).

Figure 3.4: Using an arrow, as in a video storyboard, to visualize motion in video (Goldman et al, 2006)

[1]http://www.microsoft.com/en-us/kinectforwindows/

3.3 Event perception

Information in the world around us comes to us in a continuous perceptual stream, yet we are able parse this stream into meaningful chunks. Research on the perceptual and attentional processes involved in segmenting continuous perceptual information into events may be directly informative for understanding how to visualize multimodal activity data in ways that can best be interpreted by human researchers. Event Segmentation Theory (EST), developed by Tversky, Zacks, and colleagues, argues that the human perceptual system spontaneously segments activity. Activity boundaries or "breakpoints" arise due to low-level perceptual cues consisting of abrupt changes in moment-to-moment levels of activity. These breakpoints are akin to contours in object recognition, but the dimensionality seems to be defined by motion, frequency and object-scene composition.

Studies on event segmentation commonly use variations on a procedure initially developed by Newtson (1973), in which participants divide a video into meaningful events by pressing a button to mark the boundaries in real-time while they watch the video. Using this procedure, Newtson saw reasonably good agreement across participants for the location of segment boundaries, a result that has subsequently been replicated in other studies (Hanson & Hirst, 1989). Significantly, Newtson reported evidence that recognition memory is better for still pictures taken from event boundary points than for non-boundary points, indicating that the reported structure of the event, as indicated by the segmentation, is reflective of the way the event is perceived and remembered. In addition, individuals who show greater agreement with their respective population's general segmentation pattern for a video of an activity tend to remember greater details of that activity later (Zacks, Speer, Vettel, & Jacoby, 2006). The reliability of the behavioral results supports the idea that identification of event boundaries is crucial to understanding events, but this has also been supported by neuroimaging studies. These studies, with subjects passively watching a video, show activity in visual motion processing areas of the brain, such as area MT, that is transiently increased at moments that the subjects later identify as segmentation points (Speer, Swallow, & Zacks, 2003).

More recent work has revealed further detail about this segmentation process. Participants reliably produce different segmentation patterns depending on the directions to produce larger or smaller segments, and a larger segment boundary ("coarse-grained") is likely to line up with a smaller segment boundary ("fine-grained"), indicating a hierarchical organization (Zacks, Tversky, & Iyer, 2001). Further work has sought to identify the basis for identifying event boundaries. In general, it appears that a combination of physical cues and prior knowledge about events and goals are responsible. Physical cues, such as visually identified movement of objects, can inform event boundaries (Zacks, 2004). However, these cues are mostly tied to fine-grained segments, and it has been hypothesized that semantic knowledge about the events, such as scripts (Schank & Abelson, 1977), may guide coarse-grained segmentation (Zacks & Tversky, 2001). Supporting this idea is the finding that event understanding is affected by inferences about actors' intentions that both can be derived from the stimuli and that known prior to watching the video (Zacks, 2004). As a consequence of these results, it seems likely that one's experiences and cultural environment probably directly modulate their identification of event boundaries and comprehension of events.

In many of these studies, there is a simplifying assumption that events tend to follow from one to another sequentially, with an event boundary marking both the end of one event and the beginning of another. However, if one thinks carefully about their understanding of events in the real world, this is rarely the case. The boundary points may be quite fuzzy and events will usually overlap or co-occur. For example, consider the event of eating a meal in a restaurant. Would the end be when the diners finish eating, when they pay the check, or when they leave the restaurant? Initial results have shown that overlap can affect event boundary perception. For example, in an area of overlap where one event is beginning and another ending, there is a bias toward identification of the boundary closer to the point where the second event is ending (Lu, Harter, & Graesser, 2009).

Similarly, experimental work has explored the relationship between these naturally occurring breakpoints and memory for events. For example, commercials inserted at breakpoints are remembered better than the same commercials inserted at non-breakpoints. Effective visual summaries might be created by applying relevance-based selection of salient moments of activity based on natural breakpoints (one of the approaches we plan to investigate). Schwan and Garsoffky (2004) argue that keeping the breakpoint structure in summaries maintains the temporal and perceptual cue distribution necessary to facilitate the reconstruction of past experience. These findings are in agreement with the notion that events are natural perceptual units of behavior and thus have important implications for memory. From the theoretical perspective of EST we can think of reinstating context as an act of "re-perceiving"; in other words, perceiving again the same structural and temporal relations used to guide attention and memory during the original performance of an activity.

Chapter 4

ChronoViz

I developed ChronoViz[1] with several goals in mind: to support analysis of video and other forms of time-coded data, to collect data about how researchers navigate this type of data, and to explore new techniques for visualization of this type of data. In this chapter, I will describe the design of ChronoViz as it relates to supporting analysis. Further discussion of how the interactive visualization capabilities support navigation and analysis is in Chapter 7.

The primary way that ChronoViz addresses the analysis bottleneck posed by collecting a large amount of heterogeneous data is to create synchronized interactive visual representations of multiple data streams. These combined visualizations primarily address the problem of *navigation*, helping researchers to do things such as explore data sets, find interesting phenomena in the data, find other instances of phenomena that have been identified, locate moments in the data that were identified during initial observation or previous analysis sessions.

For the research efforts that have used ChronoViz, video is usually the most import type of data for analysis. I designed much of ChronoViz with the goal of supporting analysis and navigation of video on its own, and one of the goals of integration of multiple sources of data is to make navigation of video easier. While several systems have been developed to support various aspects of this analysis challenge, ChronoViz is unique in focusing on navigation of multiple diverse data sources. For

[1]ChronoViz was known as DataPrism until October 2010, but a request from the owners of a trademark on the name DataPrism prompted the change.

example, although numerous systems exist for coding and annotation of video, such as ELAN (Wittenburg et al., 2006), VCode (Hagedorn et al., 2008), and Diver (Pea et al., 2004), they are either not designed for analysis or visualization of multiple types of data, or they do not support easily extensible visualization and navigation facilities.

ChronoViz allows researchers to visualize time-based data from multiple sources, navigate this data in flexible ways, and manually or automatically code the data with structured or unstructured text-based annotations. The data sources can include multiple video files, audio files, computer logs, sensor readings, paper notes, and transcriptions. Since these recordings may be created by a variety of devices, they can be difficult to synchronize. ChronoViz offers interactive mechanisms for aligning a wide range of data streams. Researchers can create annotations, define and evolve category structures for annotations, and construct visualization filters. Annotations can also be created automatically by employing an analysis plugin framework that supports custom scripts.

The design of ChronoViz was informed by three sources. First, previous research, drawing on both the interactive visualization research and graphical perception research described in Section 3. Second, observations and interactions with researchers. Third, analysis of interaction logs recorded through earlier versions of ChronoViz.

In the following sections, I describe the general design of ChronoViz and the design and implementation of specific features that support research workflows. Discussion of how participant observation in research groups lead to some of these design elements is found in Section 5.1.2, and discussion of how the design of interactive visualization in ChronoViz supports specific patterns of navigation is found in Chapter 6.

4.1 Design

ChronoViz began as a tool that recorded annotations about a single video, and over the course of its development, gradually expanded to supporte relatively unconstrained data sets. These data sets can include multiple individual data streams whose only qualification is that they are time-based. This can include multiple streams of every type of supported data (e.g., multiple videos, multiple sensor streams), and these streams can have different frame rates or frequencies. To accomplish this in a smooth manner, I used innovative software design layered on top of several powerful code frameworks. In this section, I will focus on four elements of the design that were critical to achieving the current capabilities: first, centrality of time; second, atomicity of temporal data; third, flexibility in visualization configuration; and fourth, loose ties between internal data representations and displayed visualizations.

4.1.1 Centrality of time

The foremost element of the design of ChronoViz is the centrality of time. ChronoViz is designed to deal exclusively with time-based data, and this restriction enables certain assumptions about how coordination and interaction with the data will occur that could not be used in a more generic visualization tool. The unit of consideration is almost always the second, and all types of data support coordination by being linked to a common clock. In every ChronoViz session, there is always a *current time*. This time is internally maintained and propagated to views of all the data sets.

This centrality of time allows visualizations to exist mostly independently from each other. There are very few assumptions made within the system about what data may exist or be visible at any given time. Even video is optional, such as in situations where ChronoViz is used to view and annotate data logs without any associated video data. This focus on time as the central element of the system enabled rapid and clean expansion of the visualization capabilities of ChronoViz. I was able to add support for new types of data, such as eye-tracking data or digital pen data, without affecting any existing functionality. The focus on time also helps the visualizations to be easily configured. Since there are very few ties between any pair of visualizations, a single visualization can be added or removed without affecting other visualizations.

4.1.2 Atomicity of temporal data

Data models in ChronoViz are divided into two high-level categories: continuous streaming media, and discrete data. Continuous streaming media, which includes video and audio, are data where the value for a given time is algorithmically determined. For example, to show the video frame for a given time, a codec will often need to consider data that is stored at several time points before and after the given time. These data are opaque to ChronoViz, in the sense that the raw data values are never directly handled, and system-provided media frameworks are relied on to generate the appropriate result for a given time.

In contrast, ChronoViz maintains low-level control over other data types. The most significant element of the design of the models for discrete data types is that every data point in ChronoViz has an associated time. This is in contrast to models that might assume a constant data rate. Such a model would be more efficient in terms of both memory and computation, since individual time values would not need to be stored and the memory location for a data point at a given time could be determined in constant time, but it would also be much less flexible. A constant data rate model would impose strict requirements on the form of data that can be imported, with little support for data streams with missing values or variable rates, such as are common in many real-world data collection efforts. Independent time values for every data point enable much of the flexibility and many of the features of ChronoViz. There seems to be almost as many types and forms of data as there are researchers, and ChronoViz has been able to support most of these data types with minimal effort.

This atomicity of temporal data also helps for supporting data streams with heterogeneous data rates. Not only does data not need to have a constant data rate, but data rates of individual streams do not need to match. With a system that required homogenous data rates among all data streams, data points would need to be interpolated, potentially generating inaccurate representations. With heterogeneous data rates, every point that is shown is one that existed in the original data.

4.1.3 Flexible visualization

One of the design decisions that the central clock enables is high flexibility in selection, configuration, and arrangement of visualizations. Visualizations are linked but independent; any one visualization can be changed without affecting another visualization. This includes both the configuration of visualization parameters as well as the subset of data that is shown in the visualization.

With visualization systems that include multiple linked visualizations, there are a number of strategies that can be used for managing the different visualizations. Among these strategies are arranging multiple visualization as panes in a single window, using a set of pre-defined visualizations and positions, allowing users to place visualizations on a free-form canvas, or having a central visualization with additional visualizations that can be layered on top. ChronoViz uses a hybrid of two strategies. Timelines are added as panes to existing visualizations, and all other visualizations exist as separate windows. This approach has a mix of advantages and limitations.

For example, one of the limitations of ChronoViz is that it is limited to working with a single document. This limitation exists for both technical and design reasons. Complex data sets, such as those that have multiple high-definition videos combined with multiple other types of data, often push the limits of the computational resources available. The decision to use multiple windows for ChronoViz visualizations presents a design challenge for working simultaneously with two documents. Distinguishing the main ChronoViz window, with the video and timelines, would be the same as any other multiple document application. But maintaining correspondences between the additional windows would be a challenge.

However, using multiple windows for visualizations has produced a number of advantages. The expansion of visualization capabilities during the development of ChronoViz, while partially enabled by the central clock, was also enabled by the multiple window design. Not only could additional visualization capabilities be added without needing to modify any existing element of the user interface, multiple instances of the same visualization could easily be supported. There are very few explicit limitations in ChronoViz with regard to the number of data sources that are supported or the number of visualization windows that can be created. Effectively, these are only limited by the resources of the computer system.

Another advantage of using multiple windows was the ease with which ChronoViz can take advantage of multiple displays. Since the visualization windows can be placed anywhere on any screen and resized to fit whatever space is available, no additional work was needed to support using more than one display. Many researchers that used ChronoViz discovered that using multiple displays was often ideal. The main

ChronoViz window containing videos and timelines is often stretched to cover most of the primary display, and windows with additional visualizations are often distributed across other displays.

4.1.4 Internal and visual representations

Essential to the design of ChronoViz is a distinction between digital representations of data and visual representations of data. This design choice gives users of ChronoViz a high degree of flexibility in choosing what data is displayed, how the data is displayed, and the arrangement of multiple data sources. It also presents some confusion to users of the system. This confusion manifested itself in several ways. First, importing data to a ChronoViz document means that it becomes available for visualization and analysis, and there is usually a default way that it is displayed. Depending on the situation, the data may not be immediately displayed or it may be displayed in an unexpected way. This introduces confusion as to whether the data is available and where it can be found.

For example, one researcher, when trying to play back an audio file that was loaded into the system but not currently displayed, asked, “Where is the audio file?” Even though the audio would be played when the document was played, the lack of visual representation of the audio file (in this case, a waveform) meant that conceptually, the audio wasn’t currently present in workspace. This is representative of confusion that exists between the visual representation of the data and the data itself.

This confusion also was found when dealing with annotations. The visual representation of an annotation on a timeline can be interpreted as the annotation itself. There were multiple occasions when I witnessed apprehension over removing a timeline over fear that removing the timeline would also delete the annotations that were shown on that timeline, erasing hours of data entry. At other times, researchers were worried that annotations had disappeared when they simply were not shown on the timelines. The flexible nature of the ChronoViz timelines, described further in Section 4.2, is a fairly unique aspect of ChronoViz. In many other video analysis systems, annotations or codes are displayed on dedicated tracks for each category, and in the absence of such clearly defined visualization, there is the potential for confusion about the presence of the data.

The trade-off for this confusion is the flexibility with which data can be displayed. With a data set that contains many sources of data and many annotations, showing all of the data all of the time is computationally prohibitive and would be visually overwhelming. The loose ties between the visualizations and the data representations allow data sets to be quickly added to the current visualization configuration, and for a single data set to be represented through multiple visualizations.

4.1.5 Implementation

ChronoViz is written using object-oriented programming techniques, and makes extensive use of the Model-View-Controller design pattern (Krasner, Pope, et al., 1988). This design pattern generally separates code into three categories: code that deals with data models, code that creates the visual user interfaces, and code that controls what data is shown and how the view responds to user interaction.

I wrote ChronoViz in Objective-C, using the Cocoa frameworks present on Apple's Mac OS X. Two frameworks in particular are heavily used. The QuickTime framework (QTKit) is used for media playback and time-keeping, and the Core Animation framework is used for most visualization. Core Animation provides capabilities that make dynamic visualization easier and more efficient, such as full-featured graphics layers, space transformations, animated transitions, and utilization of graphics processor units for increased efficiency.

In practice, the "clock" of ChronoViz is a QuickTime video object. This is useful because videos already need to maintain a time, be able to playback at different rates, and jump to a specified time. These are the key requirements for the clock in ChronoViz (and in fact make up the basic elements of data that I recorded from researchers), so rather than inventing a new component to accomplish time keeping, I can make use of existing functionality. For data sets that do not include a video, a placeholder video can be dynamically generated that matches the duration of the data.

Using videos as a clock is also useful because it is more computationally efficient to let videos control their own playback rather than being tied to an external clock. Media frameworks on modern desktop operating systems typically are well engineered and have been through years of evolution, but they are highly tuned for the playback of individual videos, and maintaining alignment between a visual data stream and its corresponding audio data stream. In ChronoViz, I align videos at the start of playback, then let them proceed individually and rely on consistent playback rates. Drift in playback is not a concern for most purposes, because video is rarely played back during analysis for a long period without any stopping.

4.2 Visualization for Navigation

There are many mechanisms for data navigation with ChronoViz, most of which are shown in shown in Figure 4.1. Since video is often a central form of data in behavioral science, the main ChronoViz interface, shown to the left of Figure 4.1, consists of a video pane, which can show multiple videos, and one or more timelines from multiple data sources arrayed below. A video that demonstrates many of the interactive features of ChronoViz is available at http://chronoviz.com/chronoviz.mp4.

Timelines are used for visualization of many different types of data within ChronoViz, and are the most commonly used interface element used to navigate the data. The timelines within ChronoViz are flexible and quickly reconfigurable. They can be used to visualize annotations, time series graphs of sensor data, audio waveforms, and individual video frames. The timelines can be rearranged by dragging them up or down, and can be resized to see more or less detail. A ChronoViz configuration can have a single timeline or as many timelines as can fit on the researcher's display. As more timelines are added, the timelines pane is automatically increased in size to maintain the same location of existing timelines, or if the timelines pane is already the full height of the window then the height of the timelines is decreased. If the minimum height of the timelines has been reached, then no more timelines can be added until one is removed.

Annotations on the timelines are displayed as either capsule-shaped marks for "point" annotations, or rectangles for "duration" annotations. The annotation marks are color-coded and aligned into rows based on the categories that have been assigned to them. The category rows can be rearranged by dragging them up or down on the timeline.

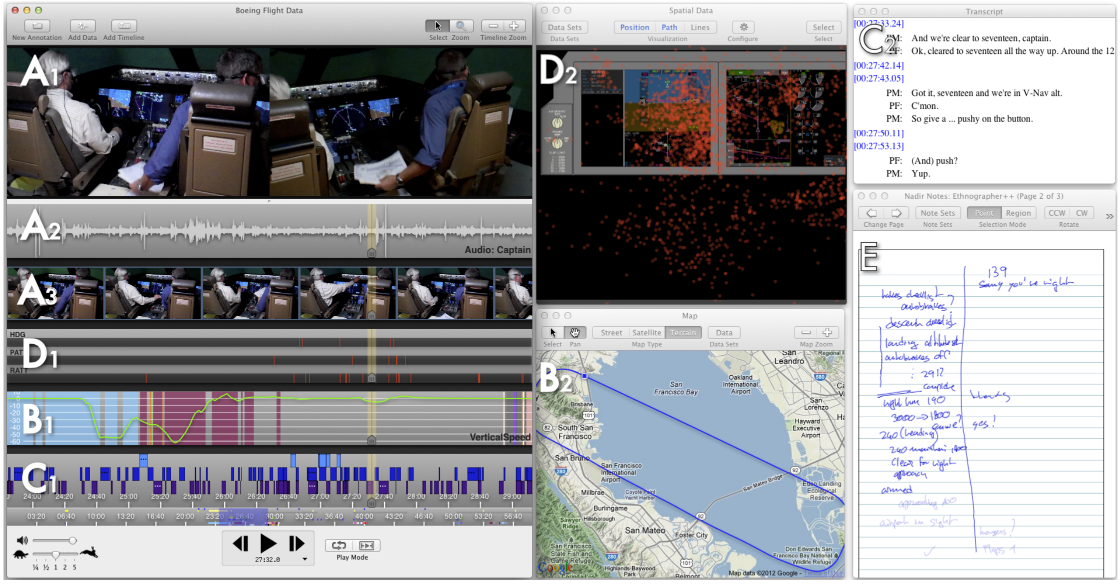


Figure 4.1: ChronoViz with flight data collected from Ed Hutchins' research, including (A1 - A3) multiple videos, also showing audio waveform and individual frames; (B1 - B2) simulator data, visualized on a map and as a time-series graph with overlaid annotations; (C1 - C2) transcript data, shown as a traditional transcript and as a timeline of color-coded annotations; (D1 - D2) eye-tracking data, shown with raw gaze overlaid on a scene camera and as gaze hits in areas-of-interest; and (E) notes taken with a digital pen.

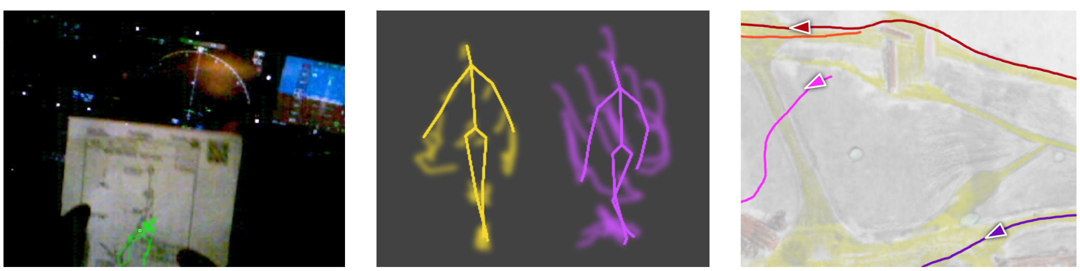
Annotations can be overlaid on each visualization, as is depicted in the time series graph in Figure 4.1. The playhead (visible as a vertical bar near the middle of each timeline in Figure 4.1) can be dragged to specify the time point shown in the videos, and can also be used to help precisely compare points on the timelines. Clicking on an annotation jumps all of the data streams directly to the beginning time of that annotation.

The alignment and reordering capabilities for the timelines and the annotations on the timelines are designed to facilitate visual identification of patterns. By making related elements of the data visually adjacent, it becomes easier to see correspondences between different types of data or relationships between different categories of annotations.

The set of annotations can be also shown as color-coded, sortable, and filterable tables. As with the other ChronoViz visualizations, table entries can be used to navigate the data. If a researcher is interested in a particular category of activity that has already been annotated, the table can be filtered to show only those annotations, and then a user can click on individual annotations to identify the corresponding data points.

Spatiotemporal data can also be visualized within ChronoViz. A special form of spatiotemporal data is geographic position data (as defined by latitude and longitude coordinates) and can be displayed on a map (currently retrieved from Google Maps), as shown in the bottom center of Figure 4.1. As with timelines, the current time position in the data can be changed by clicking on a route location shown on the map. This allows the researcher to navigate the data by navigating a depiction of physical space that is time-linked to all of the other data representations.

Generalized spatiotemporal data can be visualized in arbitrary spaces. Some examples of the types of spatiotemporal data that have been visualized with ChronoViz are shown in Figure 4.2, including gaze position as recorded with eye-tracking glasses, motion tracking data as recorded with a Microsoft Kinect, and trajectory data as recorded with a custom digital pen and interactive paper prototype. These examples all use the same underlying structures and visualization capabilities, only with different configuration.



(a) (b) (c)

Figure 4.2: Spatiotemporal visualization in ChronoViz: (a) Gaze data overlaid on video recorded from eye-tracking glasses, (b) Motion tracking data of two dancers recorded with a Microsoft Kinect, and (c) Elephant movement trajectories entered with a digital pen (image enhanced for print legibility).

With both geographic and generalized spatiotemporal data, multiple data sets can be overlaid on each other as long as their coordinate spaces are compatible. This allows researchers to compare the movement or gaze positions of multiple people. A key challenge with visualization of spatiotemporal data is representation of both the spatial and temporal components on a two-dimensional computer display. This challenge is discussed in Section 7.1.

4.3 Categorization and Filtering

To be flexible across a range of analysis and coding styles, ChronoViz supports a flexible scheme for categorizing annotations. Color-coded categories can be defined for each data set, and an optional two-level hierarchy is supported. Researchers can assign annotations a category when the annotation is created or during later review of the annotations. Multiple categories can be assigned to an annotation. This allows for flexible use of the category system. In addition to supporting traditional behavior coding, such as classifying instances of behavior into mutually exclusive categories, the ChronoViz category system can also be used to add information to annotations such as the person who created the annotation, the data that generated the annotation, or to combine information about multiple dimensions of coding.

In addition to the category system, ChronoViz also supports tagging annotations with arbitrary keywords. In some research situations, defining a category system may be too constraining or prematurely limit the scope of analysis. In these situations, keywords can be created simply by typing the relevant keyword into the Annotation Inspector. This eliminates the need to go to a separate window to define the category and doesn't lead to additional effort such as selecting a color or shortcut key. ChronoViz supports consistency of keyword use across annotations through auto-completion, or displaying a list of existing keywords as the user begins to type.

Both categories and keywords can be used to filter annotations on the timelines. Filters can be defined either by selecting combinations of categories, or by typing text to search for across the keywords or all the text fields of the annotations. As the filters are adjusted, ChronoViz dynamically changes the display of the annotations on the timeline to match the filter.

A frequent source of confusion with earlier versions of ChronoViz was filters for the timelines. Setting filters is a sporadic activity, and so I did not want to have the filtering interface be visible all of the time, since that would clutter and add complexity to the timeline interfaces.

Initially, filters were assigned using a floating window. This proved to be confusing when researchers worked with multiple timelines, because it was unclear which timeline would be affected when adjusting the filters. An initial fix for this problem was to include labels on the timeline and in the window, but a better solution was to have the filter be attached to the main window, with an arrow pointing to the affected timeline, as shown in Figure 4.3. This solution maintains a visual link between the timeline and the configuration window, even if other visualization changes are made before the configuration window is closed.

Figure 4.3: Interface for editing timeline filters.

4.4 Recording annotations

ChronoViz is designed to allow researchers to flexibly create and organize annotations about the data. There are three ways to manually add new annotations:

Annotation Inspector A window in ChronoViz called the “Annotation Inspector” (shown in Figure 4.4a) can be used to edit the properties of a new annotation. When a user clicks the “New Annotation” button in the toolbar or selects “New Annotation…” from the File menu, a new annotation is created at the currently shown time, and the Annotation Inspector is brought up to enter information about that annotation.

Annotation Quick Entry Often, editing the full information about an annotation is not needed, and a quicker method for entering annotations has a better fit in a researcher's workflow. By pressing the *Return* key while navigating a video with the timeline, the *Annotation Quick Entry* window pops up (as shown in Figure 4.4b). This window appears overlaid on the main ChronoViz window, at the current time point on the timeline. With this window, a researcher can select a category and enter the annotation text. If this annotation entry technique is used while the video is playing, the video will momentarily pause while the annotation is being entered, then resume playback when the researcher is done.

Category Shortcut Keys In certain coding situations, coders will be looking through video to identify instances of particular classes of activity. In these cases, shortcut keys can be defined for each category, and then pressing the key will enter an annotation with that category (with no additional text) at the current time. Holding down the key will create a "duration" annotation, starting when the key was pressed and ending when the key is released.

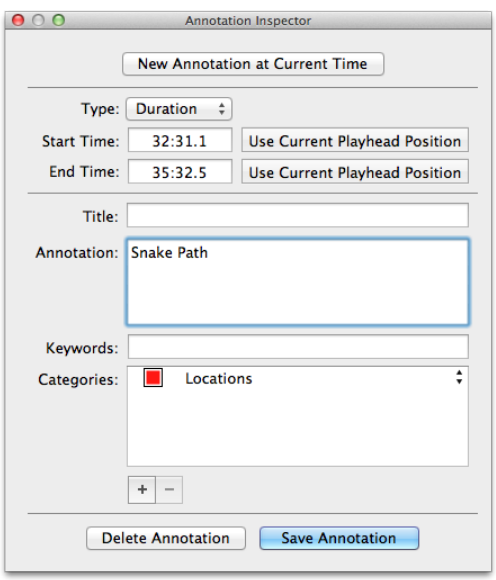
4.5 Connecting with Other Tools

A goal with ChronoViz was to make it fit within existing research workflows. Coupled with this goal was a realization that existing tools supported some tasks in ways that would be impractical to do within ChronoViz. ChronoViz supports bringing data in from several formats, has an extensible architecture to support additional formats, and supports exporting data in several formats. The most common format used with ChronoViz is Comma Separated Value (CSV). CSV is a text-based tabular data file format where individual records are separated with commas. Most data types can be imported and exported as CSV, and many other programs support exporting their data in the form of CSV data files, or formats that are easily covered to CSV. This makes CSV a great general-purpose file format.

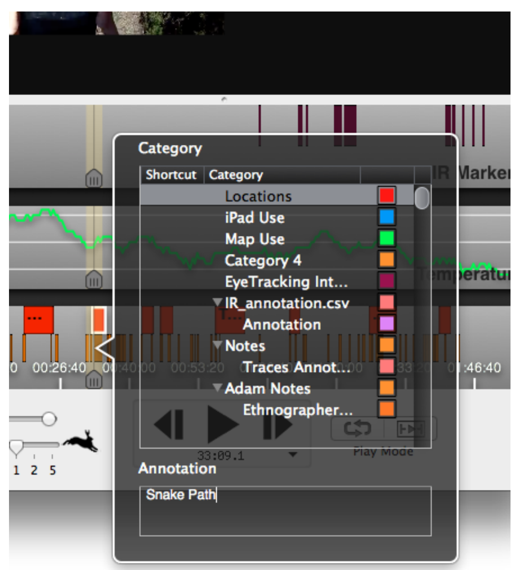
There is also support for specific formats. For example, many good tools support manual audio speech transcription, and the structure of ChronoViz was poorly suited for supporting this task in as good a way as the other tools. Several researchers that began to use ChronoViz were already using audio transcription tools. Rather than require them to switch away from using other tools, they could continue to use these tools and then use ChronoViz to create additional levels of annotation, to visualize the transcript in other ways, or to bring the transcription data together with other data. Specifically, ChronoViz supports importing transcripts that are created with the InqScribe transcription tool [2].

In addition, there are mechanisms to connect ChronoViz to other interactive tools. A custom URL scheme was developed for low-bandwidth communication with other applications. This URL scheme (taking the form `chronoviz://command`) allows simple commands to be send to ChronoViz for navigation from any application that supports clickable hyperlinks. For example, this permits linking a cell in a Microsoft Excel workbook to a specific time point in a ChronoViz document.

[2]http://www.inqscribe.com



(a)



(b)

Figure 4.4: Entering annotations in ChronoViz with the (a) Annotation Inspector or (b) Annotation Quick Entry pop-up window.

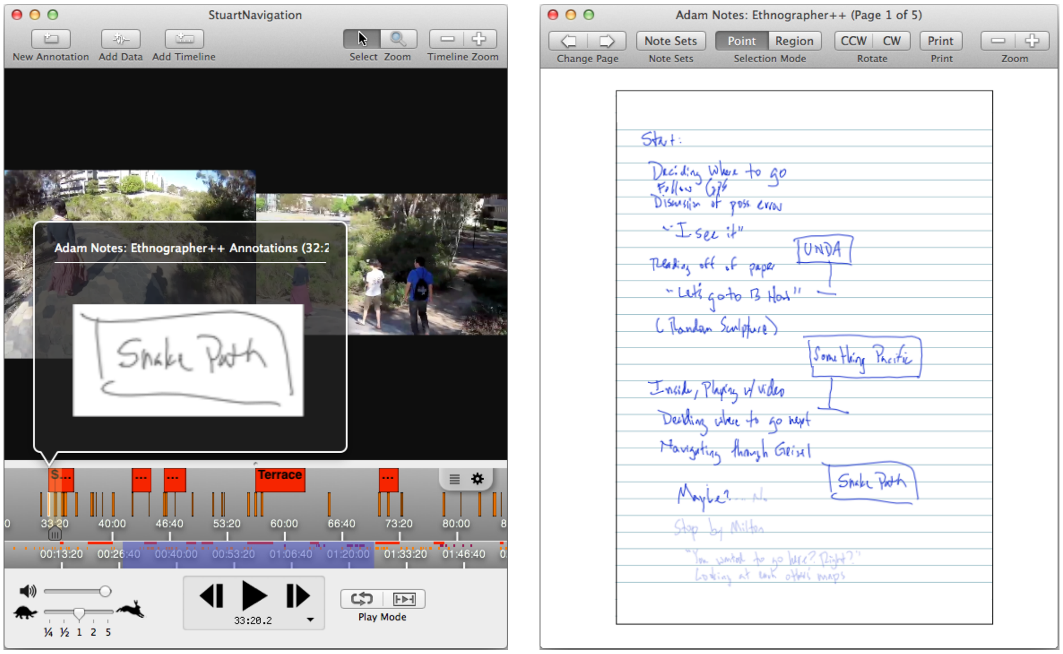
4.6 Digital Pen and Interactive Paper

A special type of data that ChronoViz supports is notes taken with a digital pen. ChronoViz exploits Anoto digital pen technology to support the integration of paper-based digital notes. This technology is based on a digital pen and a unique dot pattern printed on standard paper. The pen includes an on-board infrared camera and tracks its position on the paper in real-time by reading the dot pattern, recording not only what is written on the paper, but also when it was written. Through integration with the rest of the types of data supported by ChronoViz, this transforms the way that note taking occurs in the field, and enables new types of real-time interaction with the data in the lab.

ChronoViz supports two types of Anoto-based digital pens. The Livescribe Pulse SmartPen provides audio recording, playback, and an OLED display. It is used in batch-mode, recording everything that is written in the pen's memory, then uploading the notes to ChronoViz after the note-taking activity has occurred. Notes are taken on standard Livescribe paper or custom-printed forms, using a custom application on the pen. Once uploaded, ChronoViz merges the notes with data from other sources. The second type of pen is a wireless Bluetooth Anoto DP-201 pen that can communicate with the computer in streaming mode to provide real-time interaction with ChronoViz. Typically, the Livescribe pen is used in the field, since it can be used independently from a desktop computer and the Anoto Bluetooth pen is used during analysis, since it allows a user to simultaneously interact with data shown on paper and on screen.

Once imported with a data set, notes can be shown in two ways. First, markers can appear on the timeline that correspond to the time a particular note was written. When the researcher hovers over the marker, an image of the note appears, as shown in Figure 4.5a. Second, the entire note document can be displayed, as shown in Figure 4.5b. Pen traces are initially displayed as semi-transparent, but as the researcher moves through the data, the letters darken to indicate notes that were taken at the associated time. As with timelines and maps, analysts can change the current time position in the data stream by clicking on locations in the notes document.

ChronoViz also supports interactive use of paper-based notes for data navigation. The notes are reprinted to allow for streaming usage with a Bluetooth Anoto pen,



(a) (b)

Figure 4.5: Digital notes in ChronoViz shown as: (a) an individual note attached to an annotation and (b) a page of notes.

because a different dot pattern needs to be used with the Bluetooth Anoto pen than the Livescribe pen. This reprinted paper version of the notes can then be used for direct navigation of the data in two ways. First, the researcher can simply tap the pen on any of the notes, and the on-screen representations will move to the corresponding time when that note was taken. This has several benefits. Because the appearance of the reprinted notes is almost identical to their original form, users find it easy to recall the original context for taking particular notes. More importantly, it eliminates the need to display the notes on-screen, so that valuable display space can be used to show more data and additional detail. Since multiple sets of notes and pens can be used simultaneously, this interface also supports collaborative analysis. For example, if data is displayed on a screen using a projector, multiple members of a research team, each with an Anoto pen, can navigate the displayed data while discussing observations by tapping their own reprinted notes with the pens.

However, tapping notes only allows jumping from point to point. Since important features of data sets might only be seen in the dynamics (e.g., moving video or movements on a map), we have developed controls that can be printed with the reprinted notes, as shown in Figure 4.6, and used to control playback. These are common video controls, including icons for stop, play, rewind, fast-forward, and fast-rewind. There is also a circle surrounding the rewind, stop, and play buttons that mimics the behavior of a jog dial, such as those that are often employed for precision video editing. By moving the pen in a circular motion either clockwise or counter-clockwise on the printed circle, the video can be moved frame-by-frame forward or backward, at user-controlled speed.

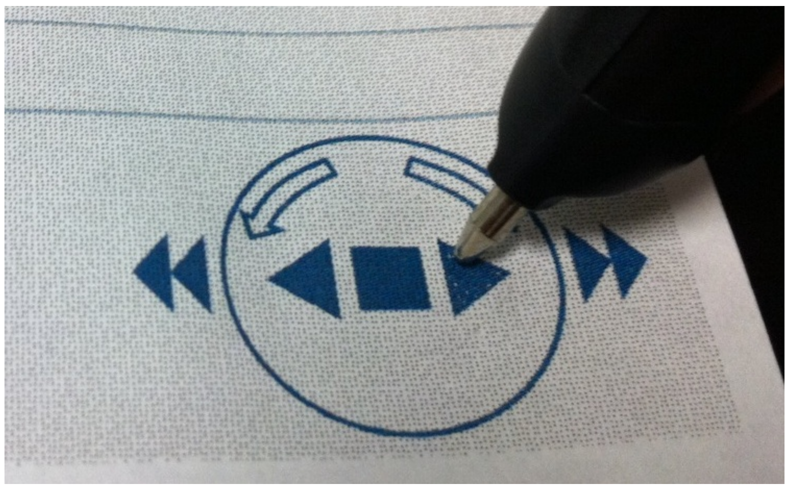


Figure 4.6: Video playback controls printed on Anoto paper. A user can tap on the controls with a Bluetooth Anoto pen to control video playback in ChronoViz.

Finally, to support playback while focusing on the notes, there are two additional techniques beyond a simple tap that can be used directly on the notes. First, by double-tapping, the time point will instead be moved to a point a customizable number of seconds before the note was written. In informal testing we found that the delay between when an observer sees an action and when that observer writes down a note often means that tapping on a note will actually move the data to a time after the described action. This shortcut allows the researcher to go to a point before the action, and then play the video to watch. The second additional interaction technique we support allows playing the video at normal speed by holding down the pen on the paper. After a 500ms delay, the video will start playing, continuing until the pen is released. This is meant to support viewing of short clips that are directly related to particular notes.

4.7 Temporal Data Alignment

4.7.1 Challenges of Data Alignment

Alignment of multiple sources of data is a challenge, and especially so when there are multiple types of data collected with multiple devices in multiple settings. This has been the case with most of the research teams that have been using ChronoViz. Each of these characteristics of typical data collection sessions for many researchers impacts the approach they use to achieve a quality alignment of the data. Consider just the use of the digital pen to collect notes. Due to the nature of the digital pen technology, it is extremely difficult to have the pen's internal clock perfectly synchronized with that of a video camera. This problem is compounded when data is collected with multiple camera, sensors, and systems

Before discussing specific challenges for achieving data alignment, I should note that almost any alignment of data will necessarily have a degree of uncertainty, and a determination for whether data are properly aligned should consist of whether an appropriate level of accuracy has been achieved. The criteria for determining this level of accuracy depend on the context of the research and the types of questions that researchers want to ask. For example, researchers that are concerned with correlating observed behavior with an EEG signal will need accuracy on the order of 10 milliseconds, while researchers that are concerned with a higher-level description of behavior in response to naturalistic phenomena may be satisfied with accuracy on the order of 200 milliseconds. The level of accuracy needed will affect the alignment techniques that are applicable to any data collection effort.

Of course, the optimal way to have aligned data is to have the data already aligned when it is recorded. This requires either a single recording device that reads data streams from different data sources (e.g., simulator data, video, and gaze position) and records them with respect to the recording device's system clock, or a signal that can be generated and be present in every data stream as they are recorded separately.

Recording to a single device might be possible for researchers that are collecting data in a consistent environment and if they are collecting fewer data streams. For example, if we were always collecting data in the same simulator and had the ability to make modifications, cameras could be installed that directed a data stream to a single computer. Even in this case, the heterogeneous nature of our data would make this a significant challenge. Gaze data, digital notes, and simulator data are all generated by stand-alone proprietary systems that would be difficult to integrate in a consolidated real-time data collection system.

For similar reasons, a central time signal is not feasible to accomplish given current technology for the data we record. It is common practice in the film industry to have a central timecode generator that is connected to the various cameras, microphones, and other recording devices. Each device records the timecode along with the data stream it is recording, and this timecode is used to align the streams during playback and editing. For this type of system to work, each recording device must be designed with connection to a timecode generator in mind. This is not the case with the devices we use to collect data, and would not be a safe assumption as we add further cutting-edge data collection devices to push our analysis capabilities.

Given that recording data that is already aligned is prohibitive in our settings, we need to develop additional techniques to align the data after it is collected. We generally formulate this problem as one of finding the temporal offsets of each data stream in relation to a timeline of the event. However, a significant assumption with this formulation is that the rate of each stream is consistent, so that the passage of time in the recorded data stream exactly matches the passage of time in the world during the activity. If exactly 30 seconds separated related actions during the actual activity, then each data stream that captures the actions should represent them as exactly 30 seconds apart. Poorly constructed data collection devices may exhibit temporal drift, so that time slowly expands or contracts in the recorded data stream. This drift needs to be dealt with before alignment or as part of the alignment process. In ChronoViz, I make an assumption that devices are tested for temporal consistency and proceed with alignment by finding offsets.

A related problem is posed by devices that capture at different frequencies. For example, one camera may record at 30 frames per second, but a simulator may only output its variables at 10 samples per second. A faster but less robust data analysis system might require the same data frequency in all data streams, so that two points are always separated by the same number of samples. ChronoViz makes no assumptions about frame rates in data, and processes the data streams to find the closest point in each data stream for a given time.

4.7.2 Alignment Techniques

ChronoViz is designed to work with many external sources of data, and treats each stream of data as an independent block of data to be placed on a conceptual timeline in relation to the other pieces of data. A ChronoViz project doesn't modify any of the data sources that are loaded; rather, those data streams will be left as is on the disk and ChronoViz will create links to those files along with metadata about the data streams. This metadata will include information about how that data stream should be aligned with the rest of the data, and ChronoViz uses this information to dynamically present aligned data to the researcher for interactive analysis.

Other visualization and analysis systems often take other approaches. For example, some systems will assume that data streams are already aligned, requiring the researcher to use external tools to align the data before loading it into the system. Other tools may combine multiple videos at different offsets and edit them into a single video file with the picture divided into multiple camera angles.

By linking to external files that can be dynamically re-positioned in time, ChronoViz allows researchers to use an interactive, iterative alignment process. A key feature of this process is that all of the data streams do not need to use the same moment as an alignment point, because alignment is transitive. That is, if data streams A and B were aligned by matching one moment, and data stream C is aligned to B by matching a different moment, then C is also aligned to A even though there was no explicit match between A and C. This transitivity of alignment makes it easier to align heterogeneous data sources that may record different aspects of activity. For example, two video streams may be aligned through a clearly visible change (such as a status light or an artificially produced visible synchronization signal such as a film clapboard). To align a simulator data stream to the video, it may not be possible to find the moment of that clearly visible change, but another event that is clearly visible in at least one of the video streams (such as an easily identifiable state change) can provide a clear signal to bring the simulator data into alignment.

ChronoViz supports a number of interactive methods for aligning data. The general pattern is to find an easily identifiable moment in the activity that can be recognized across data sources, and to find when that moment occurs in each data source. Some researchers will create this moment, such as by using a film clapboard that can be seen and heard across video and audio recording devices. Once this moment has been selected, ChronoViz gives a number of options for aligning the data, based on how the data is being viewed.

For videos, alignment can occur by clicking an "Adjust Alignment" button that reveals a slider control to move the video independently of the rest of the data. For other forms of data, the user can right-click on the part of the data that matches the moment shown in the other data sources, and select "Align To Playhead", as shown in Figure 4.7. For example, an accelerometer data set could show periods of movement and inactivity. The user could find a moment in a video when a study participant begins to move, and then click on the corresponding point in a line graph of the accelerometer data. Finally, if offsets for each data set are known, then they can manually be entered into the "Linked Files" window, which shows a list of all of the linked data sets and their start time relative to the main video of the data set.

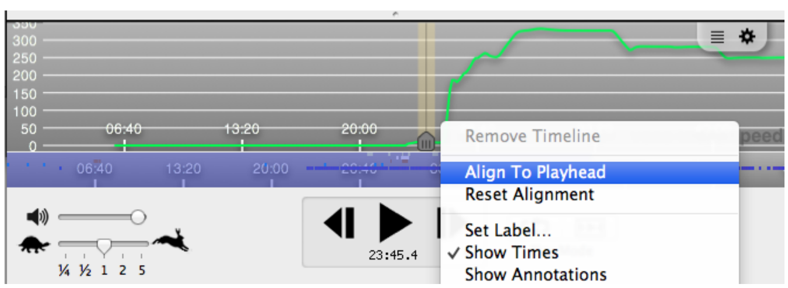


Figure 4.7: Alignment of time-series data by clicking on a point on the visualized time-series line graph and selecting "Align to Playhead".

These techniques also work for the digital notes data. The user can click on a particular note that matches up with a point in the video to set the alignment, as shown in 4.8. With the digital notes data, ideal alignment can have different interpretations for different contexts. For example, for most note-taking activity, there will be a delay between when an activity is observed and when it is recorded, due to the cognitive activity needed to perceive and interpret activity, and then translate that interpretation to words on the page. While human observers are quite versatile data recorders, we are not especially precise when it comes to recording times as compared to digital recording devices. As a result of this delay, an ideal alignment for data analysis will often not be a perfect synchronization of when the note was written during the activity. An intuitive way for the notes to be used for analysis is to click on a note to see the activity that the note was written about, but if the notes are aligned to when they were written, the video would go to a point in time just after the activity. A better alignment is to line up the notes with the activity that prompted the note to be written (or even a second or two before).

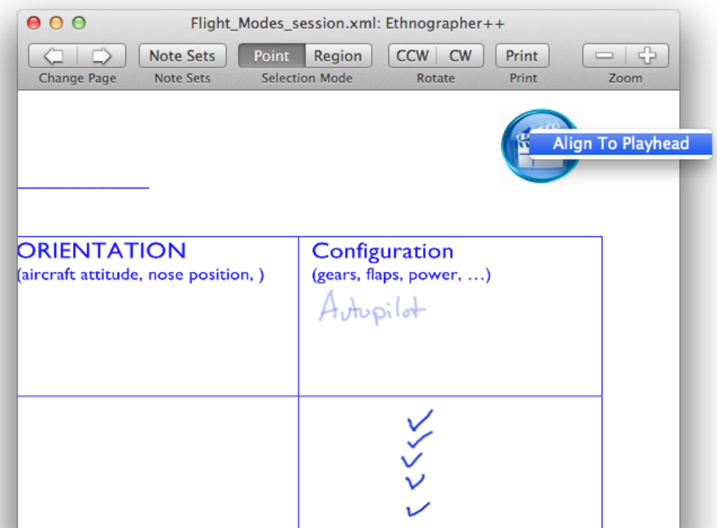


Figure 4.8: Alignment of digital notes by selecting a mark made with the digital pen. In this case, a mark was made in a specially designed “alignment” region.

4.8 Extension

In creating ChronoViz, there was tension between creating facilities that were designed for specific instances of data, and creating facilities that were generalizable to broad categories of data. Visualizations that are designed for specific instances of data can be more effective, since the characteristics of the visualization can be tailored to meet the characteristic of the data. Since they are so tailored to the data, they are also much less generalizable, which means that they can be used by a much smaller audience. Most of the visualization facilities in ChronoViz were designed with generalizability in mind, taking care to consider multiple similar types of data from different sources.

The solution used in ChronoViz to bridge the divide between visualizations designed to be specific versus general is to create a generalizable visualization facilities, and provide means for extension. In this way, the basic capabilities can be used for many domains, but specific needs can also be met. One example of this can be seen in the aviation work of Ed Hutchins. Basic time-series visualization capabilities are very useful for using flight parameters to navigate data, such as using a graph of the altitude to find a takeoff. However, more sophisticated analysis of the flight data requires more than visual inspection, and some moments of flight are determined by a combination of factors that would require looking across several variables. To address both of these needs, Professor Hutchins wrote an analysis plugin to compute additional values based on the raw flight parameters, and assign these to annotations of different phases of flight that are computationally determined by looking across several variables in the simulator data file.

Another example can be seen in a project studying middle school students solving math problems done by Sharon Oviatt in collaboration with Nadir Weibel. In this project, the students use digital pens to solve the problems, creating a digital record of all of their writing. For this project, the researchers used digital pen recordings primarily as data to be analyzed, rather than as a tool for navigation. As part of the analysis, the researchers wanted to have data about how much writing different students were doing at different times. Rather than add this feature to ChronoViz, Dr. Weibel wrote a plugin to perform analysis of the basic notes data already in ChronoViz and generate additional annotations.

4.8.1 Analysis Plugins

In addition to supporting analysis by manual inspection of the data and recording coded annotations, as described in Section 4.4, ChronoViz also has features for performing automated analysis. I developed an analysis plug-in framework for ChronoViz that uses scripts written in the Python programming language[3]. These scripts have access to all of the data that is stored within a ChronoViz file, and can generate new annotations or new data sets as a result of their processing.

The Python programming language was chosen to reduce barriers for researchers that want to create their own scripts but have limited programming experience. It is a language that is increasingly used in research environments, and also one that does not need to be compiled. Python scripts that conform to the ChronoViz plugin structure can simply be placed into a ChronoViz plugins folder, and they will appear under the ChronoViz *Analysis* menu the next time that ChronoViz is opened.

ChronoViz provides a basic user interface for the plugins, allowing plugin authors to specify data sets and parameters that can be interactively set when the plugins are run. This allows the plugin to run against specific subset of data and to create plugins that are more general, where the parameters are set according to the characteristics of the current data set. It also supports basic "what if?" testing, where the results of repeated runs of plugins with different parameters can be visualized and compared.

Use cases for analysis plugins, in the context of how they can be used to support navigation activity, are described in Section 6.3.2. Further technical details about the plugin system, including an example plugin script, are provided in Appendix A.

4.8.2 Visualization Plugins

Finally, throughout the development of ChronoViz, researchers have requested specific capabilities, and visualizations that are specific to their data. The existing visualization capabilities are designed to be rather general and apply across a broad range of research settings. While this is a powerful and productive approach, as evidenced by the existing documented use, there are often times when specific visualization capabilities are required that are not built in to ChronoViz. It would be unreasonable for visualizations that address these specific needs to be added to ChronoViz, both because there would be a bottleneck on development time, and it would add additional code and complexity to ChronoViz that would only be used by a small number of users.

[3]http://www.python.org

The solution to this problem was the creation of a more full-featured plug-in framework that supports adding new visualizations, new data models, and new file interpreters to ChronoViz. With this framework, new data types can be associated with new visualizations, so they behave the same as built-in data types and visualizations. For example, when new data types are imported, they will automatically be shown with the associated visualization, and the data sets will be selectable under the "View" menu. This plug-in framework requires plugins to be written in Objective-C, because it has more links within the ChronoViz code base.

This plug-in framework also further incorporates the design goal of integrating ChronoViz with other analysis and visualization toolkits. For example, the Hutchins-lead research team has a collaboration with Professor Janet Wiles, from the Univesity of Queensland. Her research group has created a conversation analysis tool called Discursis (Angus, Watson, Smith, Gallois, & Wiles, 2012) for doing content analysis and visualization of conversation. The conversation visualization is based on the recurrence plot (Eckmann, Kamphorst, & Ruelle, 1987) style visualization, and shows recurrence of conversation topics.

Rather than recreate this visualization, it was more efficient to link the existing visualization with ChronoViz. I created a plugin that wraps the web-based visualization with some additional JavaScript to link interactions on both sides. Through this plugin, the visualization can be shown within ChronoViz and linked with other data, as shown in Figure 4.9. Sample code from this plugin can be found in Appendix A. As with other data visualizations in ChronoViz, navigation through another type of data is reflected in the Discursis view, and navigation by clicking on elements in the Discursis view are reflected in the rest of the ChronoViz visualizations.

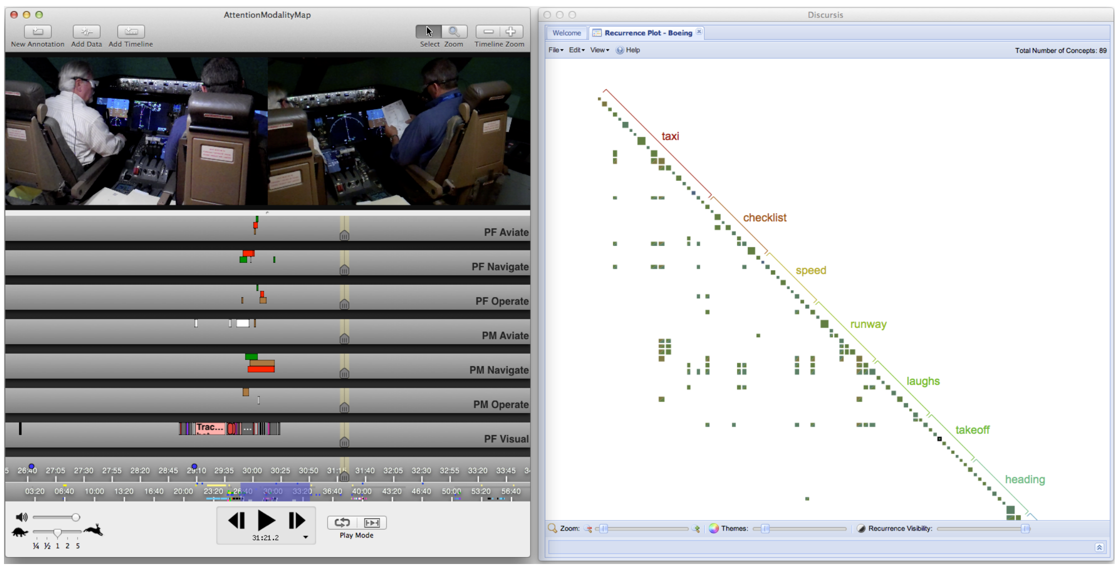


Figure 4.9: Discursis visualization of conversation analysis (right), shown within a ChronoViz window and linked to data in an existing ChronoViz session (left).

Chapter 5

Data-Driven Design

Much of the design of ChronoViz was informed by interaction with researchers that perform video (and other temporal data) analysis, and data collected from researchers while they performed analysis. The features described in the previous chapter were highly influenced from participatory involvement in a number of research projects. Further design of visualizations were influenced by data that was collected from researchers as they performed analysis.

In this chapter, I first describe my involvement with two research groups, and how that involvement influenced the design of ChronoViz. Next, I describe that data that I collected from a larger group of researchers, how I analyzed that data, and some characteristics and high-level trends of the data. In the next chapter, I discuss in detail some of the specific patterns of activity that are in the data, how those patterns are currently supported, and future work suggested by the data.

5.1 Participant Observation

To understand the process of collecting and analyzing activity of this kind, I worked closely with two research groups, described below. This involvement lead to a better understanding of how a tool like ChronoViz fits into the research process, which tasks it can support, and how those tasks should be supported. I was able to witness use of ChronoViz as researchers used it to analyze data, seeing challenges they faced due to complexities in the data, technical limitations, or poor design. The design of ChronoViz rapidly evolved as a result of these interactions.

It also lead to a better understanding of how ChronoViz can affect the decisions of which data to collect and how to collect that data. Any data collection effort is influenced by the technology that is available for recording and analysis. New technology enables new data collection techniques, but desire for new data collection abilities can also drive the creation of new technology. New data collection techniques require new analysis capabilities, but the availability of new analysis capabilities may also influence how data is collected.

5.1.1 Research Groups

I worked closely with these two research groups, although primarily with the aviation human factors research group. I was a participant-observer of the aviation human factors group, actively participating in the data collection and analysis efforts while also observing how ChronoViz was used. I had close interaction with the animal cognition research group as well, but little direct involvement in data collection and analysis.

Aviation Human Factors

In this setting, researchers are studying commercial airliner pilots by collecting rich data in flight simulators. Professor Ed Hutchins at the University of California, San Diego, leads this research, with a focus on answering questions about the allocation of attention and crew coordination as it relates to interaction with automated systems on the flight deck. As part of this research, rich multimodal data is being collected, which includes multiple angles of high-definition video, simulator logs, eye-tracking data, and notes recorded with digital pens.

Most of my observed research of this group was of a project focused on the study of the allocation and coordination of multimodal, multi-person attention. A two-person crew is responsible for flying most commercial airliner flights, and flying a modern airliner requires interaction with sophisticated and complex flight automation systems. Understanding the allocation of multimodal attention in such an environment requires a rich set of data, to understand where the airplane is in space, how the airplane is moving, how the automated systems are configured, how the status of the automated systems is displayed, where the pilots are looking, what the pilots are saying, and how the pilots are moving. One coding scheme used for analysis of this data involved categorizing activity according to three dimensions: the pilot who performed the activity, the modality of the activity (e.g., verbal or manual), and the function of the activity (e.g., navigation or aviation).

During my participation with this group, I was involved with almost all phases of the research. I helped with initial phases of low-fidelity in-lab pilot testing of capabilities, and with data collection in the field at the simulation facility of a major United States airline and the engineering facilities of Boeing. For most of the data collection efforts of this project, I assisted in the planning phases, the actual data collection, processing and alignment of the data, presentation of the data, and publication of the data.

The development of ChronoViz and my involvement with this project are intertwined, in the sense that many features in ChronoViz were at least partially inspired by observation or discussion with members of the research group. However, while some design choices were the result of specific needs of this project, most major design decisions were informed at least at some level by interaction with other groups.

Animal Cognition

In this setting, researchers are studying non-human cognitive behavior. Dr. Christine Johnson at the University of California, San Diego, leads a team of researchers, focused on answering questions about distributed cognitive activity as can be answered through analysis of visible social movement. This team includes multiple graduate students and undergraduate students, and includes study of elephant and dolphin social behavior.

Two of the studies involved analysis of video of captive female elephants at the San Diego Zoo Safari Park. These studies used the same corpus of data, with two angles of high-definition video, with one wide-angle overview video and one video focused on tracking individuals. The first study involved analysis of food negotiation events, and the second study involved analysis of movement within their habitat. For the second study, we worked closely with the research team to create a system for recording trajectories of the elephants by drawing them on paper with digital pens.

While I had much less direct participation in this group than with the aviation research group, my interaction with this group's members still had significant influence on my understanding of research process and how to best support different aspects of data collection and analysis. There were significant differences between the approach of the two groups. The animal cognition research involved much larger collections of video, and required more focused and comprehensive coding efforts.

5.1.2 Informed Design

Many of the insights for the interaction design and design of the visualizations in ChronoViz came from participation and observation of these two groups, as well as other groups with which I had lesser levels of involvement. Especially in the case of the aviation human factors research group, participation in the research process allowed me to have a greater understanding of the challenges of data collection, processing, analysis, and presentation when the complexity of data sets increases. In addition, as development of ChronoViz progressed and the user base expanded to include additional researchers and domains, discussions with other researchers gave me insight into how a potential feature to specifically support aviation research could be generalized to support a wider range of data types. In this section I provide examples of how participation and observation lead to design.

Spatial selection

Selection of notes and spatial regions (described in detail in Section 7.1) came from observations of the changing nature of note-taking in response to the digital note capabilities of ChronoViz. As note-takers began to understand the capabilities of the digital pen, notes transitioned from traditional observation notes taken in a mostly linear fashion down a page to spatially arranged notes often using ad-hoc structure to organize categories of observations (Weibel et al., 2012). Initial designs for interaction with on-screen digital notes made an assumption that notes would generally be in chronological order and without significant overlap on the page. When these assumptions are true, a user can flip through the notes (either on-screen or on paper) and understand the basic chronological relationship of the notes and the data. When this is not true, additional support is needed. The solution I created made use of a spatial query form of drill-down interaction. By selecting regions of the notes, the researcher is asking for more information about the notes in that region. This additional information is shown as annotations on a timeline. The annotations can further be explored by hovering the mouse over the annotations to show the associated note, as shown in Figure 5.1.

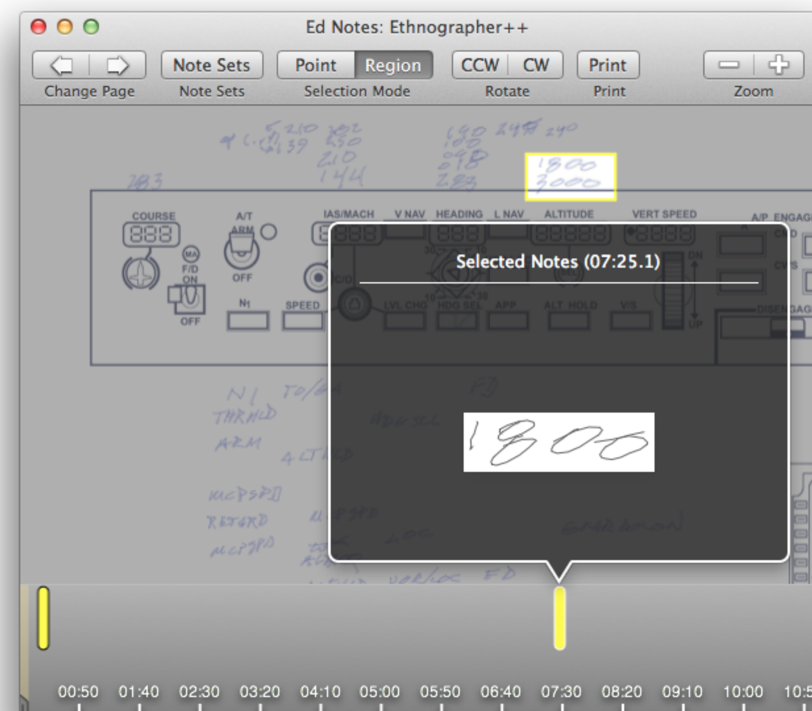


Figure 5.1: Three levels of inspection with notes. The notes are represented as marks on a page, annotations on a timeline, and an image in the pop-up window.

This spatial selection technique was further developed in response to work with eye-tracking systems. Many eye tracking systems offer the ability to specify “Areas of

Interest" (AOIs), such as particular on-screen elements or objects in a scene. This makes it possible to do analysis about when gaze was directed at those areas. There are some clear similarities to spatial selection of digital notes data: it allows transformation of spatial-temporal data to non-spatial time-series representing whether a study participant was looking at a semantically meaningful region. However, when brought into ChronoViz, the data generated about AOIs by the eye-tracking system providers was static and lacked much of the dynamic interaction capabilities of the rest of the data in ChronoViz. By expanding the dynamic selection capabilities to gaze data in ChronoViz, it also offers the ability to rapidly change the regions, to have those regions generate data that can easily be incorporated into a complex data set, and to use the capability to ask new questions rather than rely on regions that were pre-defined.

Generalized spatial visualization

Initial explorations of spatial data visualization in ChronoViz were limited to visualization of geographic data. These were primarily inspired by data from flight simulators and from GPS trackers. For example, while data from a flight simulation exercise generally will only produce a single geographic path, a study involving multiple participants would generate multiple GPS logs. This inspired the ability to have geographic visualization with multiple paths that could be dynamically added or removed, as well as the ability to independently specify whether the visualization included the full path or only the current position.

As I interacted with a wider range of researchers, it became clear that a more generalized forms of spatial visualization would be useful, and could build upon the geographic visualization capabilities. This insight came from involvement across multiple projects, including efforts to study medical interactions by collecting movement data using a Microsoft Kinect, recording gaze data in flight deck environments, and recording trajectories of elephants. Visualizing all of these data sets involves translating data coordinates to screen coordinates based on a specified coordinate space and managing multiple related data sets. By considering the needs of these three different settings, a single visualization facility was created that could support each data type, but also give researchers in each setting a high degree of flexibility for configuration of the visualizations. For example, the ChronoViz spatial data visualizations let user specify the amount of data shown in relation to the current time point. In some settings, such as the eye tracking and movement settings, it can be useful to show a short time period (usually on the order of 1 - 5 seconds) leading up to the current time point. This helps to convey the speed and direction of movement. In other settings, such as when visualizing trajectories, it may be useful to show the entire path. In yet other settings, such as different analyses involving eye tracking data, it may be helpful to show an accumulation of unconnected data points as time progresses. Creating a single visualization with such varying capabilities came about as a direct result of interaction with the different groups.

Color-Mapped Time Series Visualization

Color-mapped visualization of time-series data came from observations of analysis that placed more emphasis on comparisons between the presence and relative value of multiple variables, rather than inspection of the behavior of a single variable. Often these are sporadic data, and a traditional line graph has some problems with representation. A traditional line graph is best for data sets that have continuously varying values, such as periodic or smoothly changing values. In these cases, a line that connects data points makes sense, because the continuous data that the sampled data set represents likely had similar behavior. However, in cases such as a binary time-series representing the presence or absence of a value (such as gaze directed toward a specific area), a line graph seems ill-suited, as shown in Figure 5.2. However, before the introduction of the color-mapped time series visualization, described further in Section 6.3.2, that was the best visualization available for this type of data in ChronoViz. Observing researchers trying to use the line graph visualization for analysis of binary data was part of the inspiration for the color-mapped visualization.

Timeline management

When observing researchers as they created multiple layers of annotations and configuring their ChronoViz workspaces to support different analysis problems, it became clear that the researchers needed more support for managing their timelines. As discussed in Section 4.1, the flexible nature of the timelines in ChronoViz creates some

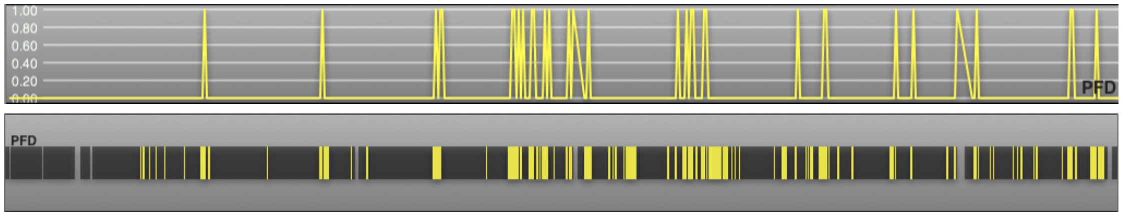


Figure 5.2: Comparison of line graph (top) and color-mapped (bottom) visualizations for binary time-series data.

conceptual difficulties, but it also creates room for creative use. For example, timelines can be arranged based on data type, keeping annotations on their own timelines and time-series data on other timelines. Alternatively, timelines can be arranged by participant, keeping all data and annotations related to that participant on a single timeline, or by concept, segregating annotations based on their conceptual relationship (which may be different from their category).

Such different uses of timelines inspired several design decisions. The ability to easily re-order timelines makes it possible to keep related timelines next to each other, and the ability to add an arbitrary number of timelines makes it possible to use the timelines to group information at different levels of granularity, such as having a large number of small timelines devoted to more specific groupings. As timelines become used for more conceptual groupings, the ability to add arbitrary labels to timelines helped to understand and remember the configuration. Originally, timelines were only labeled automatically based on the data that they contained, but this strategy fails for more complex configurations. Another design inspired by complex timeline configurations is the ability to have timelines that dynamically update based on full text search terms. A timeline that has a set of search terms associated with it can approximate a conceptual grouping if the set of annotations with those terms is close to the set of annotations that the researcher would associate with that concept. Finally, the ability to save the configured state of a ChronoViz document was largely inspired by the degree to which timelines would be configured and customized. Recreating a complex set of timeline configurations appeared to take more effort than recreating the configuration of other visualizations.

5.2 Data Collection

While participant observation gave me good insight into characteristics of the research process that influenced the design of ChronoViz, I also collected a large corpus of data about analysis activity to better understand how researchers navigate data and how this navigation should be supported. The type of research being considered often includes the collection of multiple streams of video, audio, sensed data, and activity codings, with a need to develop a rich understanding of particular activity through detailed analysis of the corpus of multimodal data. The goal was to record detailed data of the researchers' analysis of data in their specific research domain, using practices that are based on their personal established practice as adapted for use with a new interactive visualization tool. Analysis of this data has generated insight into how patterns of navigation can be supported, as well as to identify patterns that could be better supported in further work.

The primary data was recorded through the use of ChronoViz. In addition to the analysis and visualization features described in detail in Chapter 4, ChronoViz records detailed logs about the activity of researchers as they use ChronoViz. These logs include the selected time point in the data at any moment, the way that data is displayed, and annotations that are created. This data allows for precise analysis of temporal navigation patterns, and placing these patterns in the context of the visualized data. Logging these interactions provides a pervasive and non-intrusive way of recording data about visualization use, in the sense that logging can always be active, has minimal impact on normal use, collects anonymized data, and is stored locally and periodically uploaded to a server that I operate without any intervention of the researchers that are using ChronoViz.

To collect this data, I worked with several research groups, listed below. I divide these groups into two categories, based on the amount of direct involvement I had in their research. The first category includes two groups of researchers with which I had frequent direct contact, and informs the majority of this research. The second category of researchers are groups that I worked with directly but had sporadic contact, or groups that communicated with me about using ChronoViz and from whom I collected data, but had no direct contact.

5.2.1 Sources of data

In addition to the research groups described in Section 5.1.1, the following research groups participated in data collection activities related to this research. Members from each group used ChronoViz and allowed records from their interactions to be submitted. Also, since ChronoViz could be freely downloaded and used, it is likely that some portion of the log files come from unknown users.

- *Human discourse research directed by Chuck Goodwin at UCLA*
  Students advised by Professor Goodwin used ChronoViz to aid coding and analysis of video data about human discourse.

- *The Wiisard Project directed by David Kirsh*
  As part of an effort to design tablet-based digital interfaces for emergency response, a group lead by David Kirsh has collected data from an emergency response drill held in San Marcos in June 2010. They collected GPS, paper-based digital notes, and high-definition video, and used ChronoViz to help with understanding the geographic movement of individuals.

- *Gamelan research at UCSD*
  A group led by Alex Khalil, a postdoctoral researcher at the Temporal Dynamics of Learning Center (TDLC) at UCSD, is studying the temporal dynamics of children in a gamelan, a Balinese musical ensemble. They collect several angles of video, along with signals from the individual notes of the percussive instruments. As part of this effort, Deborah Forster (a researcher with the TDLC) used ChronoViz to perform video analysis to complement signal analysis of audio signals.

- *Interactive surfaces research directed by Jim Hollan at UCSD*
  Professor Hollan, Nadir Weibel (a post-doctoral researcher) and Anne Marie Piper (a UCSD graduate, now a professor at Northwestern) used ChronoViz to aid analysis of videos collected during use of digital pen technology to assist communication in medical and therapeutic settings.

- *Multisensory sensemaking research directed by Nan Renner*
  Nan Renner, a graduate student in Cognitive Science at UCSD, has used Chrono-

Viz to aid analysis of data collected about children interacting with museum exhibits.

- *Media Computing Group at RWTH Aachen*
  Students advised by Jan Borchers have used ChronoViz for analysis of video during human-computer interaction studies.

- *BAR Lab at University of British Columbia*
  Students working with Alan Kingstone and Walter Bischof have used ChronoViz for video analysis.

5.2.2 Supplementary data collection

In addition to logged interactions from ChronoViz, I personally observed many hours of video analysis, and acted as a participant observer in the research group of Ed Hutchins. In this role, I assisted with the research process as well as observed how the members of the research team used ChronoViz and other tools to analyze the data. These observations and interactions have given me valuable insight into the process of analyzing time-coded data, and grounded interpretation of the log file data as well as generating new ideas for supporting navigation.

For some of the observations, I also recorded video of researchers using the tool on desktop computers, in the form of digital video shot from an over-the-shoulder position. When recorded from a suitable angle, such as shown in Figure 5.3, live video that captures the screen, the researchers' physical movements, and the researchers' speech will allow for a richer account of the analysis activity than the logs alone. These videos are lined up with the log files, and visualized together in ChronoViz, allowing for each source of data to inform interpretation of the other.

Previous studies of interactive visualization have indicated that reference to data points across dimensions may be a crucial mechanism for understanding multi-dimensional data (Inselberg & Dimsdale, 1990; Shneiderman, 1996). This could be accomplished both through explicit computational support, such as "brushing" (where one can highlight data points in one view and have corresponding points highlighted in another view) or physical or temporal alignment of visualization scales, but also through ad-hoc methods such as physical gestures in relation to a visualization (e.g., pointing with a finger toward a particular data point on the screen) or virtual deictic gestures (e.g., indicating a data point through cursor movement). Because of the variety of types of interaction that might be used to understand visualized multimodal activity data, the availability of digital video records of some analysis sessions is a helpful resource for clarification and elaboration of the logged interaction data.

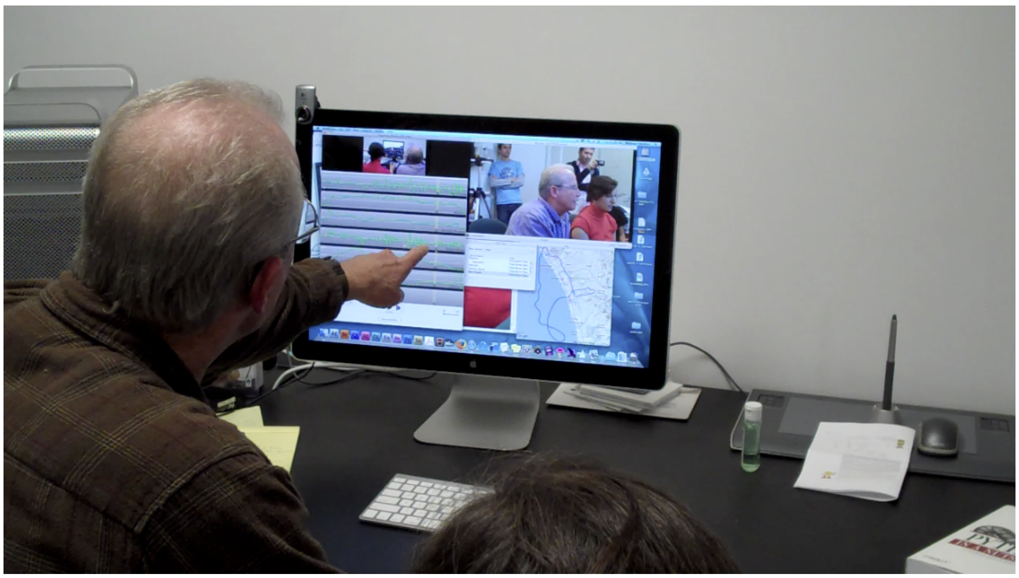


Figure 5.3: A frame from a video of Professor Ed Hutchins using ChronoViz for analysis, showing arrangement of windows on screen and a pointing gesture in relation to the displayed data.

Finally, subjective feedback was also solicited, from direct communication with researchers that participated in this study. Informal feedback was solicited from researchers as they used ChronoViz, and occasional semi-structured interviews were conducted.

5.3 Data Analysis

The collected data was analyzed in several ways, looking at individual actions and descriptive statistics at the level of individual logs files, individual videos, individual researchers, and the entire corpus. First, the aggregate logs were brought together, so that trends could be summarized, individual logs could be visualized, and patterns could be identified.

An enormous corpus of log file data was collected from 60 users, with 4328 log files that are recorded from over 257 days worth of ChronoViz activity. Since log files are created every time that ChronoViz is opened, there are many occasions when log files were uploaded that are not generally meaningful for analysis, such as when a potential user is testing functionality. I eliminated all logs that didn't include any interactions or that were for 10-second video files, which is the length of the default "blank" video that is loaded when ChronoViz is opened.

To manage this corpus, I created a special tool specifically designed to enable organization, processing, and visualization of the log files, shown in Figure 5.4. Through this tool I can visualize the log files, process the log files to determine characteristics of different types of activity, highlight these activities on the visualized log files, and combine results from multiple log files or the entire corpus.

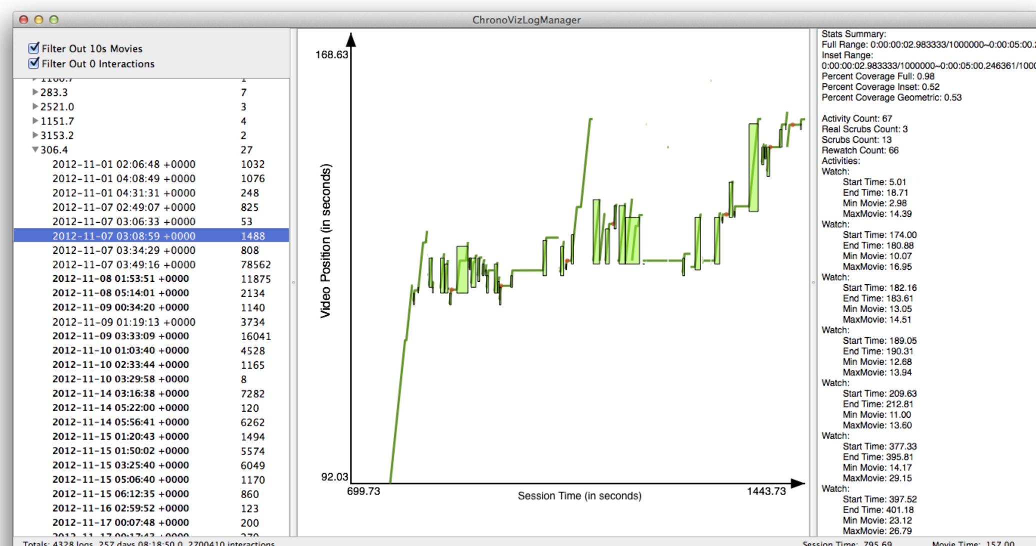


Figure 5.4: Software tool created to support analysis of ChronoViz interaction logs. On the left is a hierarchical list of logs, arranged by user then by video. In the middle is a plot of a selected log file. On the right is a summary of various measures that are computed from the log file.

The log files are recorded as a set of low level navigation events. There are four basic elements: a change in playback rate, a jump to a new time point, creation of an annotation, and editing of an annotation. Each element includes data for the time during the session when the action occurred, the playhead position, and the source of the action (e.g., clicking on a timeline or the play button). However, these basic elements need to be processed to enable further meaningful inspection. For example, there is no explicit representation of a “scrub” in the log files, but rather a series of sequential jumps to new time points.

In addition, there is a large amount of time that is contained within the log files where no analyzable activity is occurring. The nature of real-world data analysis means that a researcher may be seated in front of a computer that has data shown in ChronoViz, but may be reading related material or talking to a collaborator. The researcher may leave the data up on the screen and walk away. In these situations, the log data is still being recorded, but any relevant activity is not reflected in the logs and not accessibly for analysis unless the session was recorded on video.

To create a more meaningful representation, I generated a second level of data from the navigation events contained in the log files. Where the log files have events that occur at points, I am interested in activities that occur over a duration of time. These are things such as watching and scrubbing. A researcher may watch a video from a start time until an end time, but this is represented in the log files as changing the playback rate to “1” at a certain time and then changing the rate to “0” when they are finished. For analysis, this is translated into a “watch” event, noting the points in the movie where he started and stopped, and the time during the analysis session when the watching occurred. This collection of activities is one I refer to as “active watching”, because it is clear from the log files that the researcher was actively engaged with the video. Creating this type of data also allowed a better estimation of relative time spent in different activities.

Logs were visualized in the form of line graphs that plotted video playhead time on the vertical axis against session time on the horizontal axis. In these graphs, watching a video straight through would be represented as a 45-degree line from the bottom left to the upper right, a paused video is represented by a horizontal line, and a jump to a different part of a video is represented as a gap in the line. The logs may be scaled independently in the vertical and horizontal directions. As such, the absolute speed of playback, as represented by the angle of the line, is not directly interpretable. More importantly, a paused video is always horizontal, a playing video is always an angled line, and a scrub appears as a dotted line (actually represented as a sequence of very short horizontal lines).

A selection from one of these graphs is shown in Figure 5.5. In this example, the session starts with the playhead paused at about 14 seconds, shown by the horizontal line at the very left of the graph. The video watcher starts by moving the playhead to 4 seconds (shown by the short green line below and to the right of the first line), then to the beginning, shown by the long green line at the bottom middle of the graph. At about one minute into the analysis session, the video watcher starts playback, and watches the video until it reaches the initial point at about 14 seconds into the video. He pauses the video and waits about 30 seconds, then drags the playhead back to the beginning (shown by a sequence of dots), and immediately watches the same segment of video again.

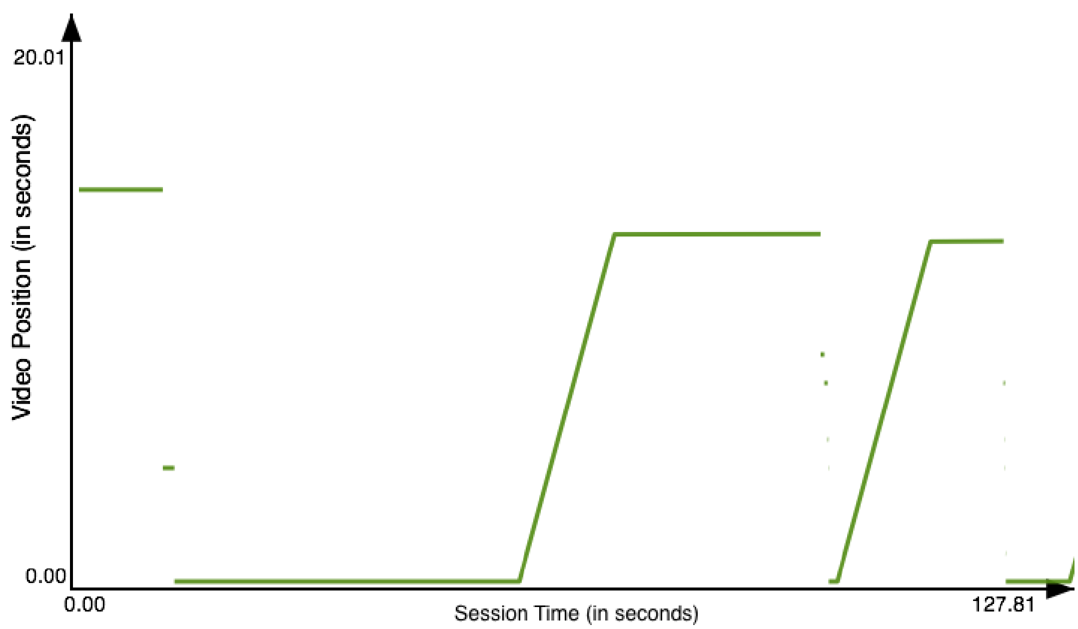


Figure 5.5: An example video activity graph, showing a researcher watching a 14-second segment of video twice over the span of about two minutes of analysis activity.

The logs contain data that let me group the logs by source and by video. Logs are actually identified by the user account on a computer, so it is impossible to be certain whether the logs from a single source came from a single researcher or multiple research assistants. There were multiple situations in the data collected where lab computers with ChronoViz were set up so that multiple users all working on the same project logged in using a single account. Nevertheless, it does group the logs by research project, and usually by researchers that at least have a similar degree of experience. Further grouping by video allows me to consider navigation patterns as they relate to different levels of familiarity with videos, and infer factors that may influence navigation. Since I did not record file names, video grouping uses the length of the video as an identifier for the video.

5.4 Data Overview

As noted in the previous section, I collected a large number of logs from analysis sessions. These logs cover a wide range of activity, both in terms of the type of research and presumed goals of the session (e.g., exploration versus coding), as well as the navigation activity that was exhibited by the researchers.

To understand some of the context of the activity reflected in the log data corpus, we can look at a few characteristics of the analysis sessions. One of these measures is the amount of the video that a researcher viewed during analysis. This gives us a clue as to one aspect of the analysis: was the researcher focused on a short segment of video, the whole video, or something in-between? This measure needs to take into account jumps from one segment to another, rather than just considering earliest and latest viewed points, because a researcher can spend an analysis session looking at a short segment at the beginning of a video and another short segment at the end of a video. To compute this measure, I divided each video into 1-second segments[1], and counted any frames that were shown from that segment. Dividing the number of segments with shown frames by the total number of segments gives a percentage for the amount of the video that was viewed during analysis.

The distribution that results from computing video coverage for each analysis

[1]Note that smaller segments could be used (down to single frames), but I make an assumption that the relatively imprecise nature of large-scale video navigation means that the results would not be significantly different.

session is shown in Figure 5.6. As can be seen in the graph, there are peaks in the distribution at the low and high ends of the scale, corresponding to analysis sessions that viewed less than 10 percent of the video, and analysis sessions that viewed more than 90 percent of the video.

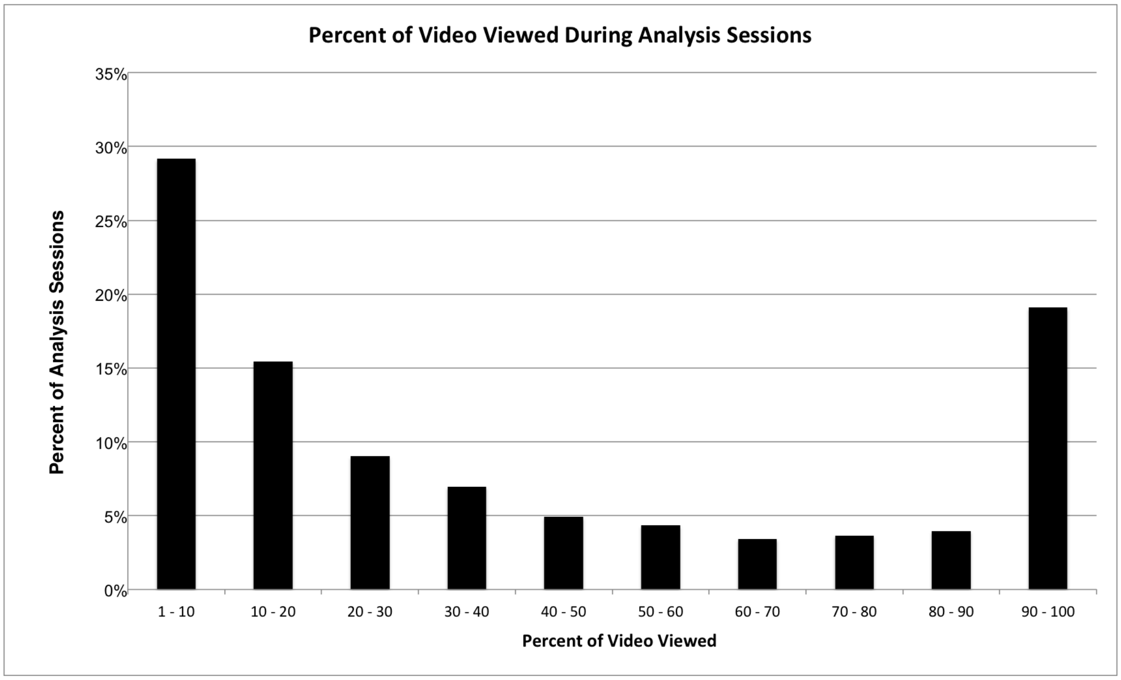


Figure 5.6: Distribution of the percentage of a video viewed analysis sessions.

A related characteristic to consider is the duration of time spent performing analysis. Simply looking at the duration of the analysis session is not a good measure. As mentioned in the previous section, an analysis session may contain long stretches of time where nothing is happening, and from the logs it is impossible to determine what a researcher was doing during that time. Instead, we can calculate the total activity time, by adding together the times of all the individual activities that are identified. The distribution of these activity times are shown in Figure 5.7. The majority of analysis sessions recorded were under two minutes (2040 out of 3727 total sessions), but since these sessions are so short they only account for about 9 percent of the recorded activity. Sessions that are under two minutes tend to have a larger number of shorter activities per minute, with an average duration of 3.2 seconds for any one activity (e.g., a single scrub or watch), as compared to 17.8 seconds for sessions longer than two minutes.

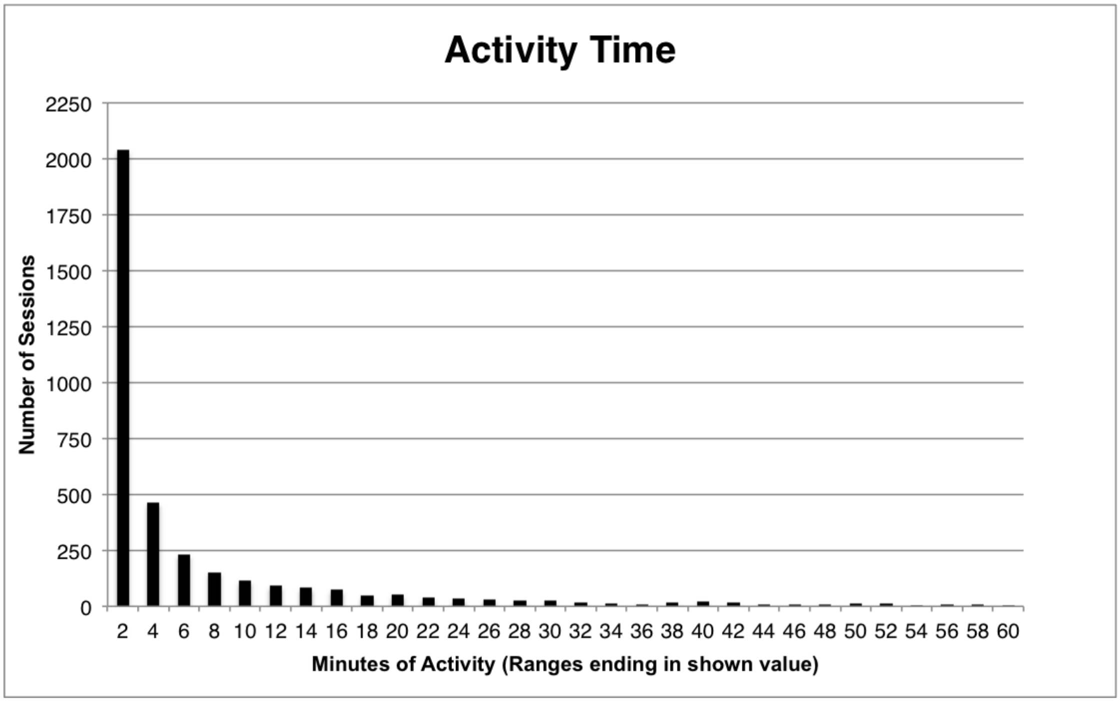


Figure 5.7: Distribution of the duration of navigation activity during analysis sessions. 53 logs with more than 60 minutes of activity are not shown.

Another characteristic to consider is the number of analysis sessions that a researcher performs for a single video. I make an assumption that this is somewhat reflective of the degree of engagement that a researcher has with that video. A video that was looked at over the course of many sessions may have had different questions that were asked about its content, or multiple layers of annotations, or multiple coding schemes applied. A video that was looked at only a small number of times may not have been closely analyzed or may have had a coding scheme applied that was already established. Figure 5.8 shows the distributions of the number of sessions for a video (per researcher). The graph is broken into three sections to show the entire data set, showing the distributions for videos that received a small number of sessions (a total of 596 videos that had 5 or fewer analysis sessions), videos that received a medium number of analysis sessions (a total of 127 videos that had 6 to 20 analysis session), and videos that received a large number of analysis sessions (a total of 36 videos with 21 or more sessions).

While the distributions presented in this chapter help us to understand the basic characteristics of the analysis sessions that produced the log file data, to understand how

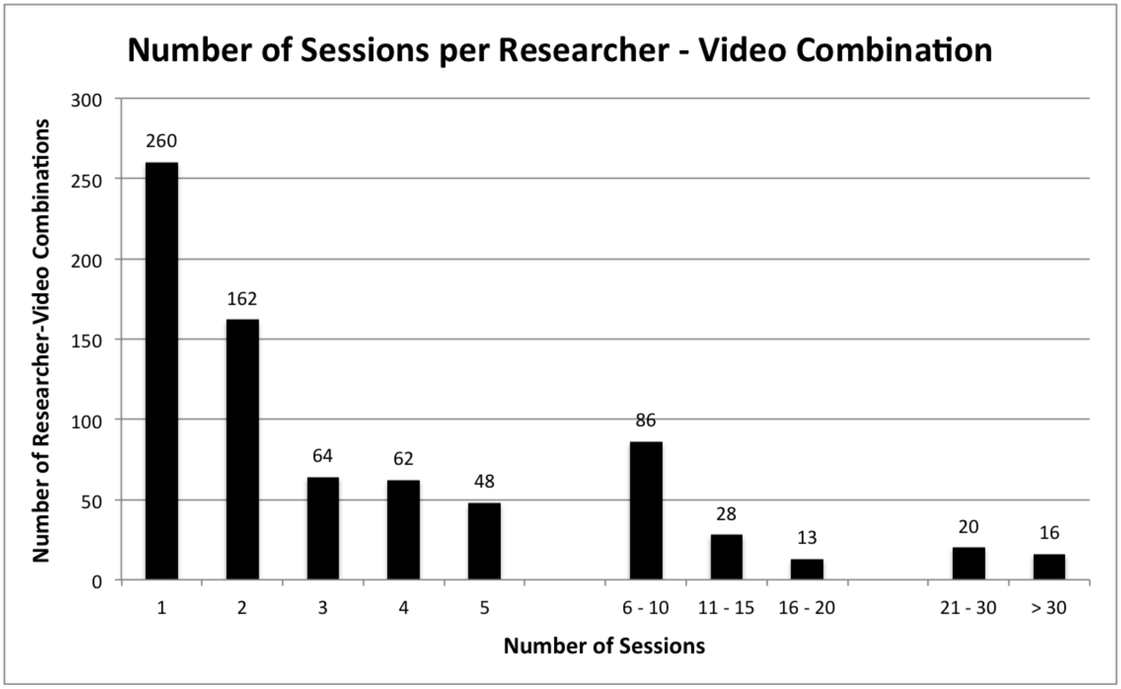


Figure 5.8: Number of sessions for each video viewed by each researcher.

the data can support design, we need to look at the individual activities in the logs. In the next chapter, I present analysis of some individual forms of activity that are present in the logs, how those forms are supported in ChronoViz, and how they might be better supported in future work.

Chapter 6

Supporting Navigation

In this chapter I discuss individual navigation patterns, and how these patterns are supported by specific interaction techniques or interface elements. These patterns come primarily from data in the log files, but are also informed by first-hand and video recorded observations of researchers performing analysis. The purpose of identifying and describing these navigation patterns is to understand how they can be supported. Following the description of each pattern is a discussion of how those patterns were supported in ChronoViz, and how they could be better supported in future work.

I divide discussion of the data by the relative temporal size of the navigation. One characteristic of the the process of video data analysis is a continual shifting between time scales. Once a corpus of data has been assembled, a researcher will first need to select individual pieces of data on which to focus his efforts, such as individual videos or collections of related data. This phase of analysis is one that is beyond the scope of the research described in this thesis. The interactive visualization techniques that are the focus of my research are primarily concerned with data that can be linked to a common timeline, and while these could potentially be adapted to help visualize a collection of videos, I did not pursue that application because they are outside of the main focus.

Once a data set has been selected, the researcher must navigate within the data to continue analysis. This navigation activity can be conceptually described as finding interesting segments of activity to look at (I call this "Inter-Segment Navigation"), and navigation within that segment itself (I call this "Intra-Segment Navigation"). This distinction is made more to structure discussion than to propose clearly defined ways to categorize the activity. Navigation activity occurs at scales from moving frame-by-frame to jumping from the beginning of a video to the end, and much navigation involves activity at multiple scales.

If we compare video interaction to reading a book, we can think of inter-segment navigation as getting to the right chapter, or back to the place where you had stopped reading the night before. Perhaps you use a table of contents and page number, or perhaps you use a bookmark. Intra-segment navigation is the actual reading of the book, usually moving from word to word, but sometimes going back to re-read a paragraph that you didn't understand, or to slowly work your way through a particularly technical or dense passage.

At a high level, watching a video straight through consists of one inter-segment navigation action – moving to the beginning of the video – and one intra-segment navigation activity – starting playback. In this case, the segment is the entire video. For actual activity during analysis, the activity is much more complex, often involving fluid movement between selecting segments of video and inspection of the frame-by-frame data.

6.1 Intra-Segment Navigation

Intra-segment navigation is most often concerned with looking at the dynamics of the activity itself. It is concerned with interacting with the video so that the movement from frame to frame can be seen and interpreted. A researcher usually needs to be able to view the video data multiple times in multiple ways to fully understand the activity that is shown. Good support for intra-segment navigation will allow researchers to easily manage their viewing of the video while giving flexibility for when the video moves, the rate of playback, and the size of movements.

For each pattern presented in this section, I begin by giving a possible definition of the pattern, then discuss some summary statistics of the presence of that pattern in the log file data. Then, I present some of the ways that the pattern is different across different parts of the data set, and discuss some of the characteristics that are seen in individual examples of the pattern.

6.1.1 Scrubbing

Characteristics of scrubbing

Scrubbing is an activity that is defined by maintaining constant control over the position of the playhead, in contrast to playback, where the user performs a single action to have the system control the playhead. A common technique for scrubbing involves holding the mouse button down on the timeline, and moving the cursor left and right to move the video back and forth at different rates. Scrubbing using the mouse cursor enables flexible control over how the video is viewed: constant movement can simulate playback, moving faster or slower can simulate fast-forward or slow-motion playback, and moving very slowly can enable frame-by-frame inspection. Multiple ways of viewing video is often a requirement for understanding activity, and scrubbing makes it possible to quickly transition from one way of viewing to another.

Another technique can use the arrow keys on the keyboard. In ChronoViz (as in many other video playback applications), the left and right arrow keys move the video forward and backward by a single frame. Holding these arrow keys down will cause the frames to repeatedly advance (or move backward), producing a kind of slow-motion playback. Some of the flexibility of moving at different speeds is lost, and more precision for frame-by-frame inspection is gained. Although the physical interface is quite different, the activity is very similar to scrubbing by using the mouse in the sense that fine-control is maintained over the position of the video, and movement is only produced by direct action by the researcher.

A final technique that I classify as scrubbing is to jump between closely placed annotations. When using the annotation table, it is possible to move between sequential annotations by using the arrow keys, which makes it possible to move through the video by jumping between points that have already been identified. Depending on the type and distribution of annotations, scrubbing in this way is less suitable for inspection of the details of movement, and more suitable for taking a larger time-scale perspective, often resulting in an effect similar to a slide show.

To analyze scrubbing in ChronoViz, the raw data of the log files needs to be processed to identify individual scrub actions. At a basic level, a scrub using the timeline is recorded as a sequence of "jumps" in the log file. However, there are a few complexities that require some additional processing. The *source* of the action also needs to be taken into account. For example, scrubbing on one timeline, then immediately scrubbing on a second timeline is conceptually two scrub actions, but is still represented in the log files as an uninterrupted sequence of jumps. Since the log files record the source of any action, this can be used to separate scrubs using different user interface elements. A second complexity is when two scrubs (from the same source) happen sequentially with no other activity in the middle. As before, this is conceptually two scrubs, but is recorded as an uninterrupted sequence of jumps. To generate appropriate scrub actions, a threshold is used for the time gap between two jumps. For the purposes of this analysis, a threshold of one minute was used, based on experience, observations, and inspection of the log files.. Note that there is not a ground truth for determining when a sequence of recorded jumps should be one scrub or more than one. A researcher can be scrubbing using the timeline, pause for 30 seconds to inspect a frame, then continue. Whether this action would be considered one scrub or two is debatable, but also would not make a critical difference for my analysis.

After computing scrubbing for all of the log files, along with other the activities described in this chapter, it can be seen that scrubbing is quite prevalent in the ChronoViz log data. It makes up 40 percent of active watching (as defined in Section 5.3), with over 204 hours of recorded scrubbing. Scrubbing shows up in most of the logs, with a scrub of at least 1 second appearing in 77 percent of all analysis sessions. The remainder of the analysis of scrubbing activity uses a lower threshold for scrub duration of 1 second.

Scrubbing activity is mostly divided between mouse-based and keyboard-based scrubbing. Scrubbing by dragging the cursor on the timeline accounts for 44 percent of scrubbing activity, and scrubbing by using the arrow keys accounts for 35 percent of scrubbing activity. Annotation-based scrubbing accounts for 10 percent, with the remaining activity performed through visualizations of data such as maps and notes.

The average length of a mouse-based scrub was 7.4 seconds, while the average length of a keyboard-based scrub was 15.7 seconds, more than double. This trend toward longer scrubbing activity with the keyboard holds when looking at the extremes, with the longest mouse-based scrub was 175 seconds, and the longest keyboard-based scrub was 1452 seconds. This difference in activity makes sense in the context of the physical aspects of the interaction as well as the types of video movement that can be achieved. Scrubbing using the mouse requires the mouse button to be held down during the entire scrub, while scrubbing using the keyboard can be either a continuous key press or a sequence of individual key presses, allowing for brief physical rest between key presses.

A somewhat different indication is provided by looking at the time span of video that is covered by a scrub. This time span is calculated by determining the time difference between the earliest point in the movie that was seen and the latest point in the movie that was seen. Mouse-based scrubbing on average covered 100 seconds of video, while keyboard-based scrubbing on average only covered 15 seconds of video.

The trend for longer average duration scrubs being performed through keyboard scrubbing and longer average coverage scrubs being performed through mouse scrubbing can be explained by looking at some of the individual patterns that are seen in the data, and some of the different situations in which they are used. Neither type of scrubbing shows a strong relationship between duration of scrub and video segment length, and both types of scrubbing can be used for both shorter segment scrubbing and longer segment scrubbing. When used to scrub over long sections of video, keyboard-based scrubbing ends up closely resembling stepwise watching (described in Section 6.1.3, in the sense that it is used to step through a significant portion of the video, stopping to make notes or interpretations. Conversely, when mouse-based scrubbing is used to scrub over long sections of video, it is often performed quickly and can allow a researcher to look for an action. This type of activity is one that fits with both intra-segment navigation and inter-segment navigation, and is described further in Section 6.3.1.

While this summary data of scrubbing activity shows some trends for how scrubs are performed, it is also informative to look at some of the differences among researchers. Although I don't have enough data to explain why different scrubbing styles are preferred by different researchers, it is clear that different situations are best suited to different techniques.

One possible advantage for scrubbing is the ability to easily move back and forth over a specific part of the video. To understand this, we can look at the amount of activity that happens within a scrub. Specifically, we can look at changes of direction within a scrub. About 49 percent of scrubs contain at least one change of direction, with a steadily decreasing number of scrubs for increasing number of direction changes, as shown in Figure 6.1.

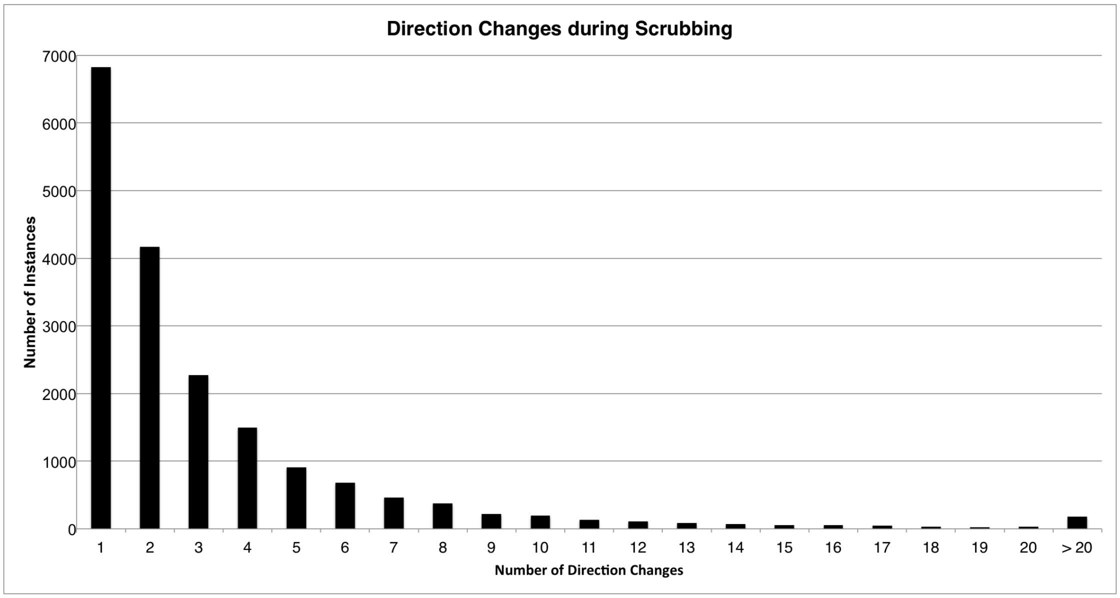


Figure 6.1: Histogram of direction changes during scrubbing.

Number of direction changes is another characteristic where differences between mouse-based and keyboard-based scrubbing appears. Many more direction changes are seen with mouse-based scrubbing. A mouse-based scrub on average has 2.5 direction changes per scrub, which translates to a direction change about every 2 seconds. A keyboard-based scrub on average has 0.8 direction changes per scrub, which translates to a direction change every 14 seconds.

This indicates that mouse-based scrubbing may be often used for rapid re-watching of a segment. Re-watching is a pattern of interaction that may be performed in several ways in addition to scrubbing and is described in more detail in Section 6.1.2.

Supporting scrubbing in ChronoViz

As can be seen in the differences between mouse-based and keyboard-based scrubbing, the trends for scrubbing styles are clearly influenced by the technology and interfaces that are used by the researchers during video analysis. Although the basic interaction styles in ChronoViz are very similar to most other video playback software, there some specific ways that some of the scrubbing activity is further supported. In addition to basic support of being able to drag the playhead along the timeline, one way that mouse-based scrubbing is supported in ChronoViz is through temporal zooming of the timelines. Zooming on the timelines changes the time-span that is shown. When controlling video through scrubbing, a zoomed-in timeline will give more precision.

Consider that the precision of moving the cursor is at most one pixel. If a timeline is 500 pixels wide (the default in ChronoViz), and a video is 15 minutes long, then the smallest movement that can be enacted through scrubbing is 1.8 seconds. If the goal is to scrub over a long time span (discussed further in Section 6.3.1), then this may not be a problem. However, if the researcher wants to perform a detailed inspection of the activity, then more precision is needed. This can be accomplished by zooming the timeline so that it is only showing the time span of the currently viewed segment. If the timeline is zoomed in so that it is only showing a 8-second segment of the video, then it takes a 2-pixel wide mouse movement to move a single frame.

Another way that scrubbing can happen in ChronoViz is while dragging an annotation on the timeline. The time for an annotation can be edited by dragging its marker on a timeline to the left or right. As the annotation marker is dragged, the video is updated, making this interaction the same as scrubbing once the marker is selected. This form of scrubbing can be useful to precisely position an annotation based on the frame that is shown in the video. The precision of this form of scrubbing can also be enhanced by considering the other annotations on the timeline. While dragging the playhead will "snap" to the time points of other annotations, to make it easier to have temporal alignment between annotations.

In addition to the timeline, scrubbing can also be accomplished through interaction with other visualizations. Specifically, the map view supports a very similar type of interaction. Rather than scrubbing over time, dragging on the map view will scrub over space. This can allow examination of activity on the basis of what was happening when participants moved from one point to another, rather than examination based on what was happening during a period of time.

One shortcoming of map-based scrubbing is that the scrubbing itself often requires more visual attention, in comparison to timeline-based scrubbing. Since timeline-based scrubbing is only sensitive to movement in the horizontal direction, mouse movements do not need to be precise, and researchers can make coarse mouse movements to scrub while their focus is on the video rather than the timeline. However, scrubbing by dragging along a map is sensitive to the precise two-dimensional position of the mouse cursor. Certain techniques are used to lessen the precision needed, such as choosing the closest point to the mouse cursor along the path, so that navigation of the video can still be achieved by relatively coarse cursor movements. But if the scrubbing is intended to help with examination of fine details of activity, then it becomes a challenging interaction for the researcher, because fine control of the rate-of-change requires spatially precise mouse movements.

6.1.2 Re-watching

Pattern

One common pattern is re-watching. Researchers will often watch, then re-watch the same segment of video multiple times within the same analysis session, often within a small time period. Often this viewing pattern results in generating or editing an annotation, so a possible interpretation of this activity is that the researcher needs to view the activity in the video multiple times to understand, clarify, interpret, or classify what he is seeing. Re-watching is an integral part of the interpretive process of video analysis, allowing the researcher to do things such as focus on different parts of the video and see activity at different time scales.

For the purposes of this analysis, re-watching is defined by viewing video that has already been viewed in that analysis session. This is a somewhat conservative definition, as it ignores re-watching that occurs over multiple analysis sessions. For example, if a segment is viewed one day and marked for further analysis, then viewed again the following day. However, without adequate data about the intended purpose of the second viewing, or even the potential for the viewing to be performed by a different person using the same computer, for the purposes of trying to understand re-watching as a form of intra-segment navigation, I only consider re-watching that happens during a single session. To compute re-watching activity, I divide each video into 1-second bins, and maintain a count of viewing for each bin. The video time span of each navigation activity is used to increment these counts, which indicates when re-watching occurs.

Re-watching is both dominant and common, accounting for much of the time of analysis activity and appearing in most analysis sessions. Of all the active viewing that was recorded in the logs, 60 percent can be classified as re-watching. Some re-watching appears in 80 percent of analysis sessions, and 46 percent of the sessions have at least half of their viewing activity devoted to re-watching.

There is a wide range of variation in re-watching style. Of the researchers that recorded more than 30 analysis sessions, the percentage of time spent re-watching varied from only 8 percent to 84 percent. The two lowest re-watchers, with 8 and 13 percent re-watching time, typically showed activity that consisted of either long sessions of stepping through videos and making annotations along the way, and shorter sessions of revisiting a few select sections. The remaining researchers were spaced relatively evenly between 38 and 84 percent of time spent re-watching.

One factor we can consider is the amount of time between one viewing of a segment and the next time that segment is viewed. Sometimes an entire video will be watched and then watched again. In this case, there will be a relatively long delay before a segment is re-watched. In other cases, a short segment will be watched and then almost immediately re-watched again, resulting in a short delay before re-watching. Of the re-watching activities recorded in the ChronoViz logs, 75 percent occurred within one minute of the previous viewing, and 53 percent occurred within 10 seconds of the previous viewing.

Consider the graph of a segment of analysis activity in Figure 6.2. At the start of this analysis segment, the researcher focuses on a segment of the video near the beginning of the video, labeled as “Rewatch Segment 1” in the figure. He views this segment four times, one time starting from the beginning, and another time continuing for about 15 seconds past the end of the segment, as defined by the eventual annotation that is made (indicated by the red dot on the graph). He then moves on to a later segment of video, labeled as “Rewatch Segment 2” in the figure. Portions of this segment are watched 13 times, with one annotation made after 3 viewings, then another annotation made at the end.

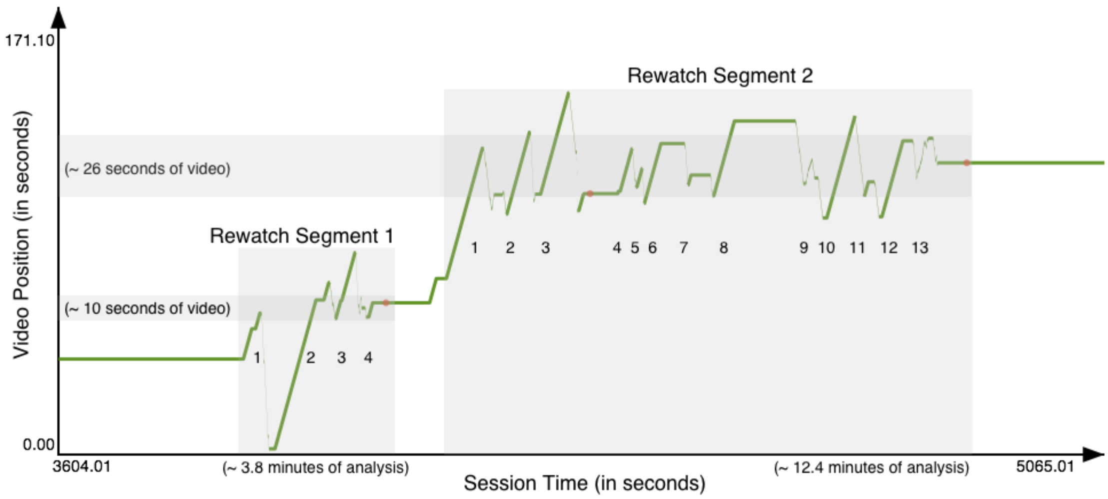


Figure 6.2: Video activity graph, showing re-watching activity for two segments of video.

Support

Several features of ChronoViz were designed to directly support this pattern. Since so much re-watching activity occurs closely in time to the previous viewing, much of the support that is specific to re-watching is designed with this immediacy in mind. For example, the configuration of the timelines and other visualizations most likely do not change between viewings. Re-watching with a longer delay is closely related to the navigation task of returning to previously identified segments, as discussed further in Section 6.3.3.

Within the main on-screen interface, there is a button to limit playback to the currently selected annotation. This option, which only works with "duration" annotations, will jump to the beginning of the annotation if the playhead is moved before the annotation, and will stop playback when it reaches the end of the annotation. By creating an annotation that lines up with the beginning and end of the segment of video, this makes it possible to re-watch a segment of video without having to pay close attention to the location of the playhead. There is also an option to loop playback (start playback from the beginning once the end is reached). Looped playback works in combination with limiting playback to selections, to make it possible to automatically play a segment of video over and over.

Another way this pattern is supported is through the digital pen interface. When a researcher taps on a printed note, then holds the pen down, the video will start playing from that time point and continue playing until the pen is lifted. If the researcher wants to re-watch that segment of video, he can simply tap the pen again in the same location. One nice side-effect of this style of interaction is that the researcher doesn't need to look at the paper to re-watch the segment, since the pen is being pressed repeatedly in the same location. Instead, the researcher's visual attention can be focused on the video.

While re-watching often happens in an immediately sequential manner, at other times researchers will watch one segment of video, then view other parts of the video, and go back to the first segment again. In this case, support for re-watching needs to be combined with support for finding your way back to the first segment. One way to do this is with tracking and visualization of one's history of activity in relation to the video, to represent where in the video the activity has been focused. Some techniques for this are discussed in Section 7.2. Although generalized activity history visualization might keep could complete history, this could be adapted to specifically support re-watching in the same session by incorporating a decay function to only represent recent segments that have been watched.

6.1.3 Stepwise Watching

Pattern

Another pattern is one I call stepwise watching. This pattern consists of watching a segment of video, stopping and possibly making an annotation, then continuing to watch the video. The segment of activity shown in Figure 6.3 is an example of this pattern. The researcher alternates watching and pausing, occasionally creating an annotation when he pauses the video.

While it is easy to understand this pattern in general, defining what is and is not a stepwise watch is actually problematic. For example, consider the activity graph shown in Figure 6.4. In this segment of analysis activity, the researcher is exhibiting navigation activity that could generally be called "stepwise watching", but with a variation. Each time the video is stopped, the researcher also does a short period of scrubbing, to move

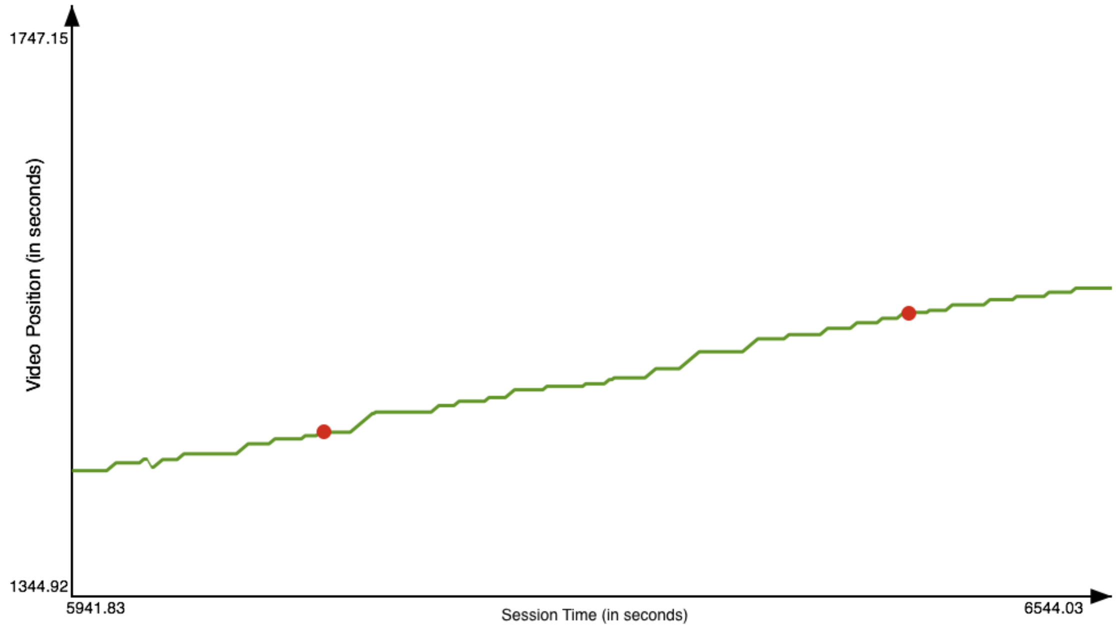


Figure 6.3: Video activity graph, showing stepwise watching activity leading to the creation of two annotations, as represented by red dots.

the playhead of the video back in time.

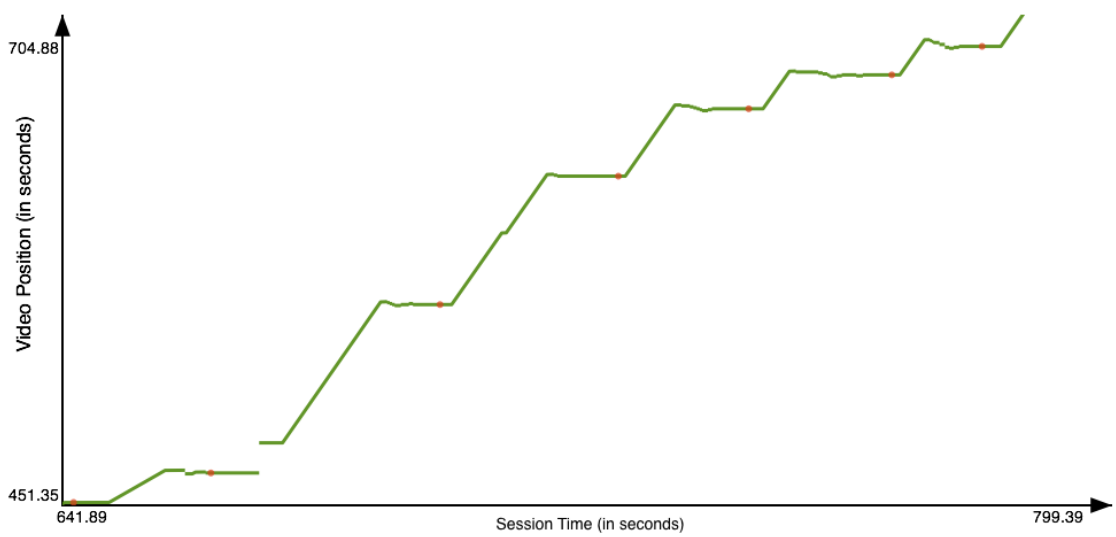


Figure 6.4: Video activity graph, showing stepwise watching activity with intermediate short scrubs.

Stepwise watching appears in 35 percent of all analysis sessions, but is more prevalent in early sessions. It appears in 52 percent of first sessions per video, and 40 percent of second and third sessions. Annotations are created about a third of the time: 33 percent of stepwise watching instances (a continuous sequence of play-stop-play) include the creation of an annotation, and an annotation is created for 37 percent of all play-stop combinations.

Support

There are two parts of this activity to support: starting and stopping playback, and entering annotations. One of the ways that I supported this activity pattern in ChronoViz is to link the two. The *quick entry* mode of entering annotations is specifically designed to enable researchers to quickly enter annotations while stepping through video. When the entry window is brought up, the video will pause if it was playing, and resume when annotation entry is finished. This mode of annotation entry is also designed to support all operations from the keyboard. When stepping through video, it is often easier to use keyboard controls (using the space bar to stop and start playback) so that visual attention can be focused on video rather than user interface elements. The *quick entry* window can be brought up by pressing the "Enter" key, the annotation can be typed, and then the window can be closed by pressing the "Enter" key again.

Another way to support this activity is to take a different perspective and consider the need to stop and restart playback. If the purpose of stopping playback is simply to record an annotation about the video, then interfaces can be designed to enter annotations without stopping. For example, assigning shortcut keys to annotation categories in ChronoViz allows minimal annotations (with a category but no other information) to be entered while the video is playing.

While there are occasions where this is the case, at other times the stopping may also function to give the researcher time to reflect on what he has seen before moving on. Translating perceived activity into concrete language seems to be an important cognitive bottleneck in some phases of observation and analysis. In this case, interaction methods to support this pattern should aim to maintain the video dynamics while minimizing the need to interact with the system.

6.1.4 Remaining challenges

One of the areas for future work is to develop more flexibility for watching at different time scales. Imagine a system that let you quickly and easily select a segment of video to focus on, and then provided specialized facilities for moving about within that segment. The temporal zooming capabilities of ChronoViz are aimed to support this style of interaction, but there is clearly room for improvement.

Early versions of ChronoViz that were limited to a single annotation timeline attempted to explicitly support this type of interaction by automatically displaying an additional timeline when an annotation was selected (see Figure 6.5). The timeline showed frames from the video and was limited to the time range of the annotation, so that the researcher could simply move the mouse up to the additional timeline and have more detailed control. However, the implementation proved to be confusing to users. It was unclear when the timeline would appear and disappear, how to interact with two timelines at different time scales, and why one timeline showed frames while the other showed annotations. In addition, once support was added for an arbitrary number of independently configured timelines, the automatic zoomed timeline was challenging to visually integrate with the other timelines, and eliminated. A better design for a similar timeline might address some of the difficulty in navigating at multiple time scales.

6.2 Creating Annotations

Although not strictly a form of temporal navigation, manual creation of annotations is often an important part of navigation and worth considering in this context. Annotations are not only an eventual result of data navigation, but also may be an important part of data navigation. They are a result in the sense that they are a record of how the data was interpreted. They may contain a description of the activity that further interpretation can be built upon. They may be categorized in such a way that quantitative analysis of the number and distribution of annotations is meaningful.

They are part of navigation in the sense that they are a form of temporal bookmark. Since interpretation of video data is an iterative process, with representations built upon representations (Roschelle & Goldman, 1991; Goodwin, 1994), annotations may

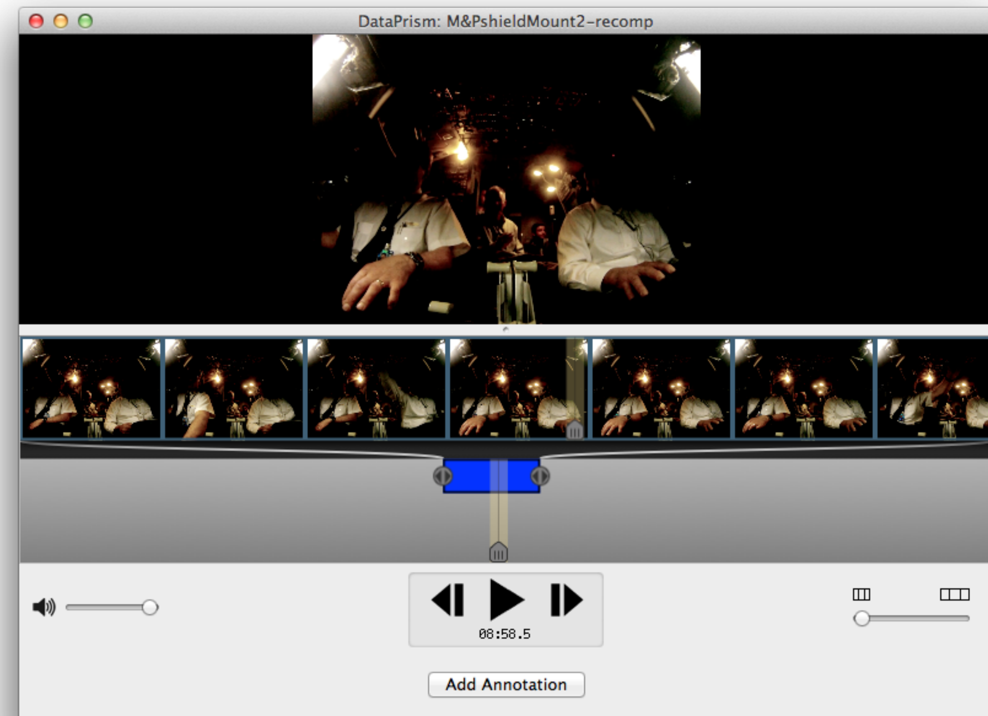


Figure 6.5: Early version of ChronoViz showing an automatically generated timeline with frames from a selected annotation.

serve as an intermediate representation. They may indicate areas where further attention is required and provide a record of existing interpretations.

The ChronoViz annotation system was designed with both of these uses in mind. Annotations can be exported for use in other programs and can also be used to specify parameters for exporting other data. For example, a common way to share insights is by showing and describing a video clip. In ChronoViz, it is easy to export a video clip that corresponds to the time range of a selected annotation, as shown in Figure 6.6. A guiding principle for the design of the interactive visualization of annotations was to support iterative refinement and accumulation of insights. To allow for this, annotations should be easily used for navigation, so that segments can be revisited, and easily edited, so insights can be refined.

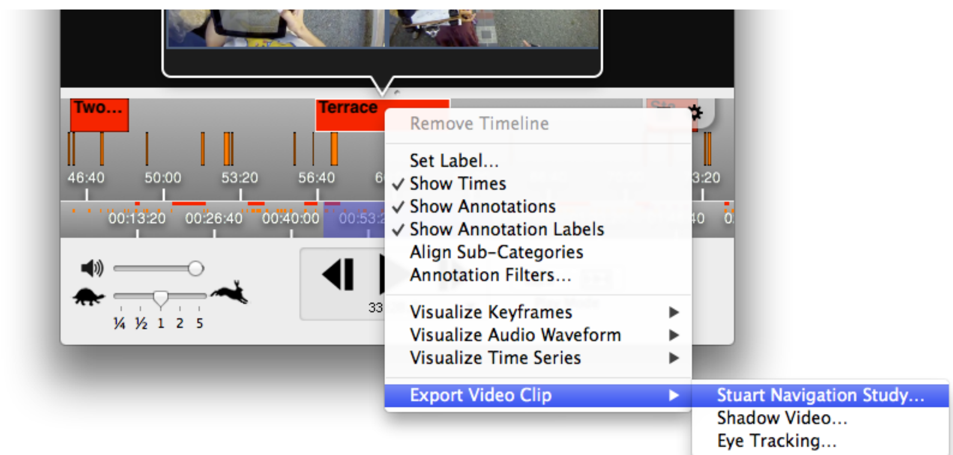


Figure 6.6: Exporting a video clip by selecting an existing annotation. The sub-menu gives the option of exporting a clip from one of three videos that are associated with the data set.

Although there are many ways that the interactive annotation editing and visualization facilities I created support these tasks, they also fall short in some ways. Observations of researchers as they created and edited annotations reveals several ways that annotation systems can be improved.

One way is to provide more flexibility with how annotations are coded. In ChronoViz, there are three primary pieces of data associated with any annotation: Time, Annotation Text, and Category. Later on in its development, I added the ability to add

Keywords. Both Categories and Keywords can be considered a form of categorical data. The distinction between Categories and Keywords is somewhat artificial, but reveals tension about the level of structure that is supported by the categorization scheme.

A potentially better way to support categorization would be to have more flexibility for moving between different levels of structure, and for evolving how the categorization scheme is defined. For example, a mechanism for assigning a category to all annotations that have a specified keyword (or combination of keywords), could support a workflow where keywords are first used to record interpretations, then the distribution of keywords helps to create a category structure, then that category structure is easily applied based on the keywords.

Better interfaces for modifying categorization schemes could also help. Frequently, I observed confusion as to how to create and edit categories. Creating easier ways to see the categories and keywords that are available, to consolidate multiple categories, or to add or remove categories from groups of annotations would make the category system more flexible by making it easier to modify an existing category structure.

Another way to improve the annotation system is to provide better support for grouping annotations and managing groups of annotations. One side of this is manual grouping. A group of annotations may be related to a particular coding pass or to a particular question, and have annotations with diverse categories. Some of the existing mechanisms of ChronoViz were used for this purpose. The ability to give an annotation multiple categories means that an annotation can be given a category that corresponds to an activity classification (e.g., a particular type of gesture) as well as a category that corresponds to a context (e.g., a phase of activity). Categories were also used to specify the source of an annotation, such as when annotation were imported from a different file. Researchers generally found using categories in this wide-ranging way to be a bit confusing, possibly because the same digital form of information might be used to represent very different types of information. Having the ability to manually create groups of annotations would help to manage visualization of the annotations, as well as to more directly support the iterative nature of video coding.

Another side of grouping is to design ways to automatically group annotations.

More sophisticated searching techniques and search interfaces could certainly be used. Better support for tracking annotation meta-data such as creation and modification dates, and who created the annotation, could also help. This would allow different analysis sessions to automatically be grouped together visually, and older annotations could be hidden if newer annotations are more relevant. Showing only recent annotations could help with navigation by making it easier to return to a segment that was annotated earlier in an analysis session.

6.3 Inter-Segment Navigation

Inter-segment navigation is generally about finding segments of activity to watch or to focus on for further detailed analysis. Two prototypical types of this navigation are exploratory activity, such as discovering segments that have not already been identified, and searching activity, such as finding or re-finding segments of activity that are already known. In the space between exploratory activity and searching activity is activity that may have aspects of both, such as in cases where a segment of activity is suspected to exist based on things such as knowledge of the structure of that class of activity, memory of perceived activity during the data collection, or impressions from initial viewings. In addition, the activity that precedes the navigation is important to consider. Navigation may occur when first looking at a video, when coming back to a video, or in the middle of an analysis session after some segments have already been analyzed.

Since researchers rarely (if ever) come to do video analysis with no existing ideas about the video, we can support navigation that uses existing beliefs as a guide for where to look. Sometimes these beliefs are more temporal (e.g., something happened about halfway through), sometimes they are sequential (e.g., something interesting happened when the participants got to the third step), and sometimes they are situated (e.g., something interesting happened when they were working on a particular part of the problem). A goal with designing support for inter-segment navigation has been to support access to data in these different ways.

6.3.1 Long time-scale scrubbing

When a researcher is searching for a particular activity or instances of a class of activity, he may move quickly over the time space of the video, watching the frames to get an overall sense of the activity. If we use 30 percent of the video as a threshold for long-scale scrubbing, then 15 percent of the scrubs in the log files are long time-scale, and all of the researchers except for one had at least one long time-scale scrub. Individual researchers had up to 20 percent of their scrubs be long time-scale scrubs.

One thing that can make this type of activity more productive is to provide additional context. The scrubbing activity does not need to be equally distributed over the course of the video. If additional cues are provided, regions that are probably inconsequential to research questions can be quickly skimmed, leaving more time to go over the remaining regions with slower, more detailed scrubbing. The same techniques that are described in the next section to aid identification of potential segments apply to providing context for scrubbing.

6.3.2 Identification of Segments

While time scrubbing is a common and straightforward way to find segments of video, one of the goals of the interactive visualizations I have created is to provide new ways of finding segments that are worth further attention. An underlying theme is to make better use of human visual search capabilities to enable different ways of searching through the data. Increased visualization is especially needed with increased data, not only in terms of raw quantity of data but also in terms of more types of data. By making it easier to see anomalies or patterns, focused analysis of the video could be more efficient. There are three techniques used in ChronoViz to aid identification of segments: visualization of time-series data, using data processing to produce annotations, and searching and filtering.

Visualization of time-series data

The goal of integrating time-series visualization with video playback is to make it possible to navigate the video by visually identifying patterns or anomalies in the time-series data. Since the whole time span of the time-series data can be seen at a high-level much easier than the video, it should be more efficient to identify a specific point along a time-series graph than by searching the visual information in the video. Depending on the characteristics of the data, it may be used akin to a table of contents, providing a good high-level outline of the course of activity, or as data worthy of inspection on its own or in the context of other data.

The characteristics of the time-series visualizations in ChronoViz were designed to support these two complementary activities: using the time-series as a navigation aid for the rest of the data, and for inspection of the time-series data itself. For further inspection of time-series data, temporal zooming can be very useful to overcome display space limitations. Even for a somewhat average data collection rate (30 Hz) and activity duration (15 minutes), the number of data points is an order of magnitude greater than can be displayed on a reasonably large display (on the order of 20,000 data points vs. 2,000 pixels).

While much time-series data is of relatively smoothly changing data or periodic data, where a traditional line graph is a good visualization, other data that researchers have used is of a more sporadic or binary nature. For example, the data from an eye tracker that is set up to detect areas-of-interest can be represented as a sequence of binary time series, indicating whether the gaze is directed at each area. When visualizing this as a line graph, it is a sequence of peaks that appears cluttered and uses much more visual space than is needed for the data. In addition, the relative scales of the data and display will often not lend themselves to easy discrimination of different densities of data. To address these problems, I created a time-series visualization that is based on changing color values. Rather than mapping data values to a vertical pixel location based on the data value, values are mapped to a color range. For a binary time series, this results in a strip of color, typically with bright stripes indicating a positive data point (as shown in Figure 6.7). This visualization is more space efficient than a line graph, so multiple time-series can be visualized in the space normally allocated for a single time-series. In the case of gaze data, this can be especially useful because the close spatial location of multiple data sets facilitates comparison between different areas.

This color-based visualization can also be useful for understanding the distri-

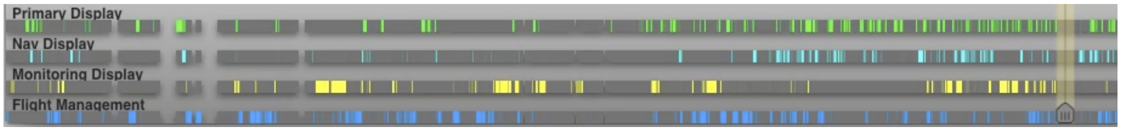


Figure 6.7: Visualization of four binary time series, showing data from four regions of gaze data from a flight simulation. Bright colors represent gaze directed at that region, dark grey represents gaze directed away from that region, and transparent represent missing data.

bution of sporadically spaced data. For example, ChronoViz supports viewing data recorded with the Microsoft SenseCam (Hodges et al., 2006), a small device with a camera and sensors that hangs from the neck. A single SenseCam data file may have data from multiple sessions, and using the color-based visualization to represent the presence or absence of data can be useful for navigating among the sessions. Figure 6.8 shows an example of this usage.

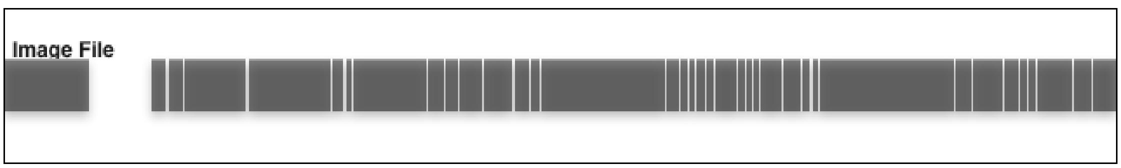


Figure 6.8: Color-mapped time-series visualization, adapted to show presence of image data to aid navigation of data recorded with a SenseCam.

Based on subjective reports from researchers using the visualizations, the color-based visualization is easier to interpret in these situations. However, this visualization was introduced relatively late in the course of this research after observing difficulties with line graph representations, so wider use would be needed to gain a better understanding of the situations in which it is helpful. Further experimental work would also be useful to help understand how different visual representations of time-series data may be better for different types of data.

The color-based visualization is likely worse for accurate interpretation of the values of data points, depending on the color scale chosen. Since interpretation relies on identification of specific color properties, as opposed to determining a spatial position (as is the case for a traditional line graph), accurate interpretation of specific values is sacrificed for relative comparisons and rapid general impressions. However, since

ChronoViz timelines can be easily reconfigured, a timeline that is displaying a time-series by using the color-based visualization can be quickly changed to use a line graph should it be needed.

Using data processing and annotations

The annotation system of ChronoViz was designed primarily to support manual entry and interaction with annotations. In real-world use, it also ended up being a valuable way to display additional types of data and the results of data processing. One way that additional data was displayed through annotations was with transcript data. While an obvious way to integrate speech transcripts with the rest of the data was to display a transcript with somewhat standard formatting and presentation (e.g., as a list of utterances labelled with speakers and times), a less obvious way was to translate the utterances to annotations. Utterances map quite readily to annotations, with the text of the utterance taking the place of the annotation text, the time of utterance mapping to the time of the annotation, and the speaker of the utterance being recorded as a color-coded category that is assigned to the annotation. In this way, the transcript becomes transformed to something that visually represents the pattern of conversation. All of the utterances from each speaker are aligned into rows and color-coded, and since the order of categories on the timelines in ChronoViz can be rearranged by dragging, relevant speakers can be arranged so that their utterances are shown on adjacent lines.

With this timeline-based visualization of transcript data, display space constraints mean that the individual words of the transcript are less visible, but that higher-level characteristics of the conversation are more visible. For example, the color pattern of the annotations will make it easy to see if there was rapid back-and-forth, if speakers dominated alternate segments, or if a single speaker dominated the conversation. For example, Figure 6.9 shows transcript data from two pilots, revealing a characteristic back-and-forth conversation pattern as one pilot asks the other to perform a task or for a piece of information, and the other pilot reports back. Since it exists on the interactive timelines of ChronoViz, the visualization created by the speech pattern can then be used to guide navigation of the video data. Also, since ChronoViz supports multiple visualizations of the same data, both the timeline and traditional view of transcript data can be used. The traditional view is linked in time to the rest of the data, so by clicking on an utterance that is represented as an annotation, the corresponding text will be shown in the traditional view.

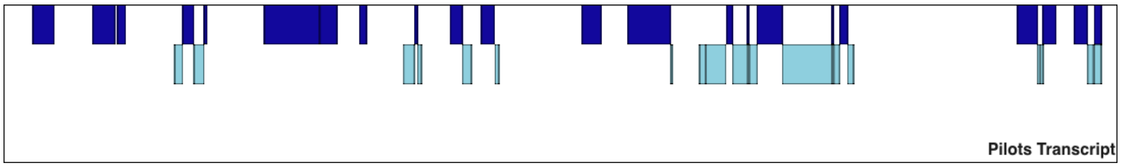


Figure 6.9: Transcript of two pilots, with utterances of each pilot shown as a row of annotations on a ChronoViz timeline.

An even more compelling use of annotation visualization for helping to identify segments is in visualizing the result of data processing. While visualization of raw data recorded from various systems and sensors is quite powerful, additional processing can often improve understanding of the data, make the patterns or transitions in the data clearer, and automate some of the tedious aspects of data analysis.

There are two patterns for visualizing the results of data processing as annotations. The first is to use external tools to generate additional data files that can be imported into ChronoViz. There are several instances where this pattern has been used, including for visualizing audio source angle as recorded from the Microsoft Kinect[1]. This data is saved as time-coded angles, representing the angle of the dominant speaker in relation to the Kinect. As a sequence of time-coded data points, the default visualization of this data in ChronoViz is as a line graph. However, if the researcher who collected the data is interested in using this data to help understand conversation (and not in the raw acoustics), then this line graph is not a very good representation. It implies a single moving source and makes it difficult to understand groupings of the data. A better representation can be generated by converting the audio angles to a format that can be imported as annotations. In this way, the representation will mimic the annotation-based transcript visualization, as seen in Figure 6.10.

The second pattern is to use ChronoViz's data processing plugin system, technical details of which are provided in Appendix A. This system is explicitly designed to generate new annotations and to use the visualization of annotations as a way of viewing

[1]http://www.microsoft.com/en-us/kinectforwindows/

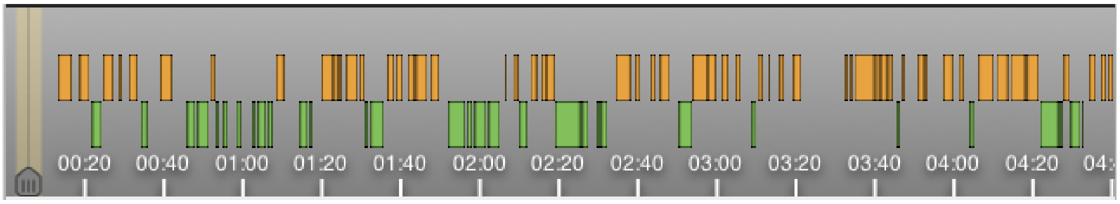


Figure 6.10: Audio angle data recorded with a Microsoft Kinect of a conversation between two people, pre-processed to create annotations to represent the conversation.

the results of data processing. Researchers in a number of domains have made use of this capability. In some situations, it has been used to generate a "first-pass" layer of annotations, giving some basic structure to the data. These plugins have used heuristic-based methods, taking advantage of known characteristics about the domain, to identify phases of activity. In other situations, plugins have been used to highlight patterns or find relationships among multiple data sources. In all of these cases, the results can be shown as colored annotations on timelines, and used to guide further watching of video or other exploration of data.

The plugins can also generate additional time-series data sets. For example, noisy data may be filtered, or summary measures can be generated from multiple data sets. These can be used to guide navigation in the same way as existing time-series data sets. For example, Figure 6.11 shows the results of a plugin created by Nadir Weibel to compute a measure of writing speed for notes recorded with a digital pen. In this case, the data is from a project lead by Sharon Oviatt, focused on middle school students solving math problems. The time-series graph computed by the plugin is overlaid on manually created annotations that indicate the different problems.

In both of these cases, the annotation system is used as a mechanism for visualization of structured and color-coded information. The visual patterns created in this way can be used to quickly see where relevant things in the data have been identified and the temporal relation among them.

Searching and Filtering

While visual search is a powerful way to inspect data, at other times text-based search may provide a better alternative. For example, when searching for specific con-

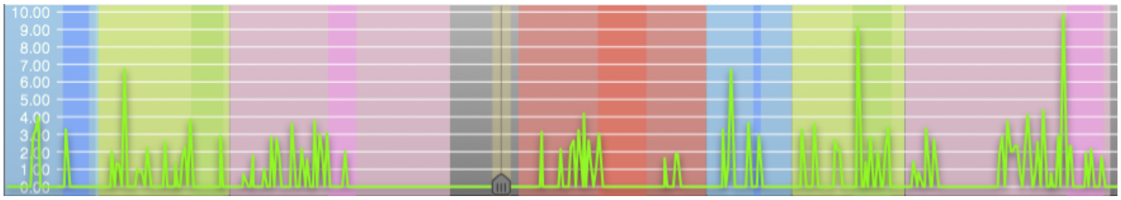


Figure 6.11: Results of running a plugin to compute a measure of average writing speed (shown as a line graph) from digital notes data.

cepts in annotations or transcript data, searching for words can help to find those time points. Text-based search fields exist in several places in ChronoViz and are integrated with the visualizations. The most traditional search is found in the transcript window (shown in Figure 6.12, where searching will scroll to and highlight the entered text if it is found in the transcript, and move sequentially through all the found locations in the transcript. Once found, the highlighted text can be clicked to move to that point in the data. Other search functions in ChronoViz serve to filter annotations. In the annotations table, for example, all of the fields of the annotations (e.g., annotation text, title, categories) can be searched by entering text in a search box at the top of the window. On the timelines, annotations can similarly be filtered. Filtering the timelines determines which annotations are shown on the timeline, so the visual pattern represents the time points as well as the categories of information at those times. These visual patterns can guide identification of new segments, as well as help return to existing segments.

6.3.3 Returning to segments

After becoming familiar with a video, further analysis will require returning to segments of video that have been previously identified as interesting or deserving of further attention. Perhaps the researcher is continuing on with a path of inquiry that he was previously exploring, or wants to shift to looking at different segments of the video. In each case, the researcher will need to find the segments before continuing.

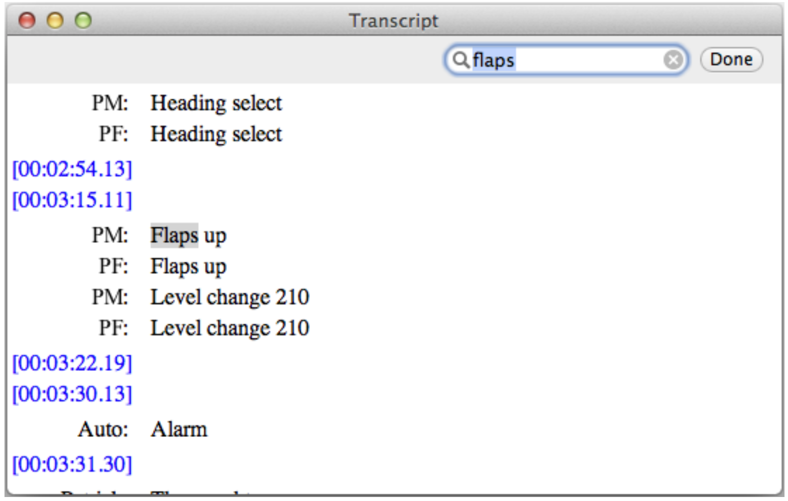


Figure 6.12: Searching in a transcript from a simulated flight for occurrences of the word "flaps".

Using state maintenance

The simplest situation is when a researcher wants to return to the same segment that he was looking at during his last analysis session. A researcher may work with a segment of video over the course of multiple days, perhaps because of time limitations, or because of simply needing time to reflect. In these cases, when the researcher returns to the video the next day, the obvious choice is to show the last time in the video that was shown when the file was closed. To support this, ChronoViz records the current time point when a file is closed, and shows that time point the next time the video is opened.

However, if the goal is to support resuming analysis of that part of a video, then the state of the visualizations can also be recorded and reinstated. For complex data sets, with multiple sources of data or extensive annotation systems, selecting and configuring how the data is displayed can be time consuming and tedious. An important part of this is configuration of the timelines. Multiple timelines may be set to display different subsets of annotations and data, and this configuration may be critical for triggering memory of where the researcher was in the analysis. When re-opening a file, the goal should be to return the state of everything that is displayed so that it looks the same as when it was last open.

In addition to returning to the state when the file was last closed, it can also be useful to record different states that correspond to different segments, different questions, or different phases of analysis. A researcher may want to have timelines configured to show a different set of annotations, or there may be a time-series that is relevant for one question but not for another. ChronoViz supports recording an arbitrary number of states, and a new saved state can be added at any time by selecting a "Saved Configurations" menu item. Once saved, the configuration can be reinstated simply by going back to that menu.

Note that this also reveals one possible area for future work: to make the selection and configuration of visualizations an easier task. For example, while the entire state of a ChronoViz analysis session can be saved and resumed, the same ability is not currently available for individual elements. Since timelines are so configurable, creating easy ways to save specific timeline configurations and add them to an existing set of timelines, or to create timeline "templates" that can be used for multiple similar data sets, could give added flexibility for configuring the state of ChronoViz to support a given analysis task.

Using annotations

The annotation system itself is designed with this purpose in mind. There are two main reasons for creating annotations: recording insights, and enabling further analysis. I designed several ways to visualize and interact with annotations specifically to support the latter purpose.

Representing annotations as clickable colored bars on a timeline makes it easier to return to segments that have previously received attention. It transforms the task from having to find a precise time to having to identify the correct annotation. If the category (and color) of annotation is known, along with a region of time, then a visual search of the timeline can narrow down the possible candidates. The text of the annotations can be read, the mouse can be hovered over the annotations to see an image from that time, or the annotations can be clicked to move from one to the next to find the desired time point.

Occasionally, researchers will choose subsets of a video on which to focus analysis. In these cases, the annotations create visual clusters at the locations on the timeline that correspond to these segments. These can be used in combination with zooming the timelines to progressively get to the time that is being searched for. If annotations are used to define the regions of time for the video subsets, then the timelines can automatically be zoomed to just that time span, enabling a faster return to the context of analysis for that region.

Using digital notes

While annotations may provide the clearest way to get back to a segment, the digital pen tools can also be used for this purpose. Even though they are easily modifiable, annotations imply a sense of commitment toward codifying interpretations of the data, and require some effort to create. In comparison, handwritten notes seem more informal, and have as much or as little structure as the writer cares to give them. Some researchers prefer to make handwritten notes to record their initial impressions.

To allow researchers to use this more informal note-taking style but still take advantage of the digital time linkages in ChronoViz, we created the ability to use bluetooth-connected digital pens for interactive digital note take during analysis. Researchers can record free form notes on paper while watching video, and the notes are reflected in an on-screen copy of the notes. The notes are instantly interactive both on the paper and on the screen, and can be used to go back to the video point that was showing when the note was recorded.

The utility of this system points to another interaction style that could be quite useful for recording impressions. In addition to discrete annotations, it could be useful to support time-synced free text entry, with text editor window that looks like a normal text editor (e.g., Notepad on Windows or TextEdit on Mac OS X), but records the time when any word is typed. An interface like this would fill a gap between free-form paper-based recording and structured annotations. Supporting a continuum of structure in this way, similar to supporting a continuum of coding structures, could help support a workflow that moves from less structured to more structured data.

Using activity tracking

Sometimes during initial viewings of a video, a researcher may not be ready to create categorized annotations or even record thoughts on paper. In these cases, automatic tracking of where the researcher was looking at in the video can be useful. Simply by looking at different points in the video, a record is created that can be used to guide the next analysis session. Visualization techniques for representing such activity histories are discussed in Section 7.2.

6.3.4 Remaining challenges

One of the shortcomings of ChronoViz with respect to inter-segment navigation is the reliance on annotations as a way to mark regions of the video. As noted earlier, the ChronoViz annotation system enforces a certain amount of structure that may make it a poor fit for recording certain kinds of data. For example, while ChronoViz annotations make effective bookmarks for segments that have been identified and can be described or classified, they don't work as well for creating simple short-term bookmarks, such as in situations where a researcher just wants to mark a time point to revisit without making any comment about what is at that time point.

A common question asked if it was possible to highlight regions of time across multiple timelines or multiple types of visualizations. Two possible complementary solutions to this problem are better mechanisms for selecting regions of time, and better ways of visualizing those selections. Selected regions of time can be useful to understand relationships between different data sets or to filter visualizations to only show data related to that segment. Currently, the only way to indicate a region of time is to create an annotation, but this can be more effort and imply a more permanent record than some exploratory activity might require. Ways to make more ephemeral time selections (that could then be converted to more permanent annotations) could enable better exploratory activity and deeper connection between linked visualizations.

Chapter 7

Visualization of Time-Coded Data

While most of the design in ChronoViz was inspired by data and observations of specific elements of current practice, as described in Chapters 5 and 6, further exploration of visualization ideas was inspired by a more general view of analysis activity. In this chapter, I discuss these additional concepts that build upon the individual visualization ideas discussed in the previous chapters. I begin with a discussion of how spatiotemporal data is supported through interactive visualization. Next, I discuss how a researcher's activity during analysis can be recorded and represented as a way of supporting navigation and reflection on their own activity. Finally, I discuss how I expanded the idea of multiple linked visualizations that is extensively used in ChronoViz to explore linking visualizations across devices.

7.1 Visualization of Spatiotemporal Data

Much of the data that researchers collect has the form of spatial data that varies over time. Examples of this include geographic position data, eye-tracking data, and digital notes data. Even video can be considered in this way, in the sense that it is a spatial array of pixels that changes over time. Visualization of data that has both a spatial nature and a temporal nature necessarily involves tradeoffs. Normal video playback privileges presentation of the real-time dynamics of the activity at the expense of presentation of an overview of the data. Frame-based representations are usually the opposite, focusing on presenting an overview of the data but presenting limited information about dynamics. This tradeoff is a general problem for visualization of spatiotemporal data, but most apparent with data that does not have a simple linear quality. For example, eye-tracking data, where the gaze position may jump from location to location, is more difficult to understand than a GPS log of a single person's position, where movement will likely be understandable from a static representation of the path and knowledge about the timing of a few key points. Consider both eye-tracking data and digital notes data. In both cases, the location of a data point (e.g., the gaze position or the position of a stroke of the pen) is not strongly predictable from the previous point or from knowledge of the activity. A researcher may have some assumptions about where they expect gaze to be directed for segments of time, and short sequences may be predictable, such as during saccades.

To overcome some of these limitations, I developed several techniques for visualization of this type of data. The first involves selection of spatial regions to reveal temporal aspects of the selected data. This technique relies on the ability to have multiple linked visualizations, specifically a representation of the space of the data, and a timeline of the temporal nature of a selected region. When a researcher selects a spatial region of the data, a timeline appears below the spatial visualization to represent the points in time when data exists in that region. Depending on the data, different representations are used on the timeline. For digital pen data, annotations are created that correspond to each group of pen strokes, as shown in Figure 7.1. For eye-tracking data, a horizontal bar is displayed with vertical colored bars representing points in time, as shown in Figure 7.2.

Many existing eye tracking visualization packages have somewhat similar features (for example, see Tobii Studio[1]), but an important advantage of my technique is the interactive and dynamic nature of the visualization. A researcher can select different regions in real time and see the resulting time-series. This capability supports both "What if?" style exploration, and also allows researchers to easily adjust for minor variations in positioning that may be introduced because of imprecision in devices or observers, such as poorly calibrated eye-tracking glasses or researchers recording observations with a digital pen outside of a specified region on their paper.

[1] http://www.tobii.com

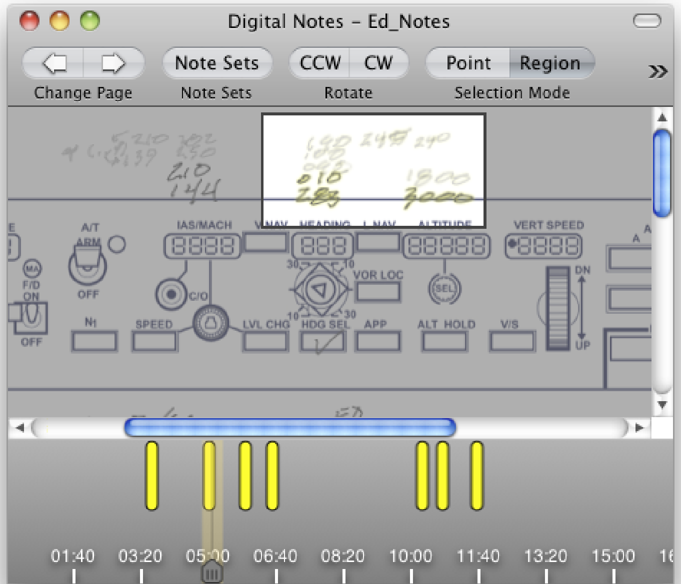


Figure 7.1: Selection of a spatial region of notes generates a timeline below the notes, showing the time points when those notes were made. In this case, notes were made by annotating a diagram of a flight instrument, and the timeline reveals times when adjustments to the auto flight settings were made through the instrument.

A further technique that builds upon the dynamic selection of regions is to show transition percentages for the selected regions, as shown in Figure 7.2. This technique provides a middle ground for understanding some aspects of the temporal nature of the data that is lost when displaying the entire set of spatial data, but without having to perform close inspection of the time-series displayed on the timeline. Arrows are used to represent the transitions, with widths that are scaled according to the relative percentages, to visualize how often the gaze moved to each of the other regions. Each region can be clicked to show just the transitions from that region, and when an annotation is selected on other ChronoViz timelines, the transitions are limited to the time range of that annotation. This allows researchers to quickly see how the transitions during one activity compare with those during another activity.

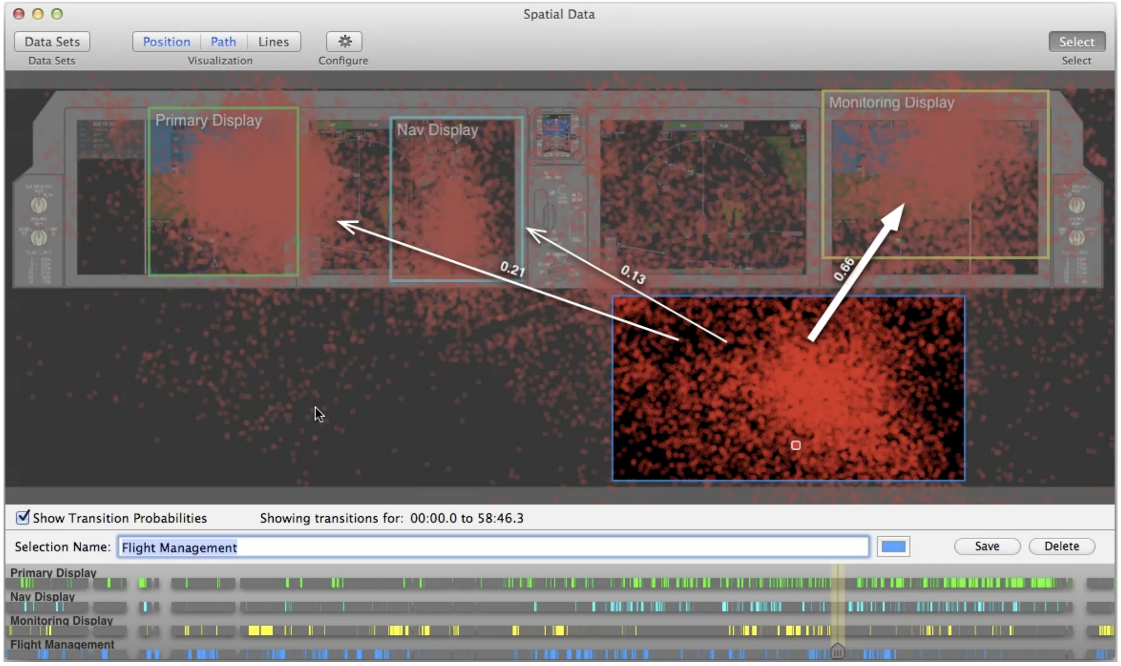


Figure 7.2: Selection of a spatial region of eye tracking data, with transition percentages shown for a selected region.

7.2 Leveraging Activity Histories

To support the early exploratory stages of video analysis, I created techniques for visualizing a researcher's history of interaction with a video. This draws inspiration from attribute-mapped scrollbars (e.g., Edit Wear and Read Wear (Hill, Hollan, Wroblewski, & McCandless, 1992)) and a conceptual model of physical wear on objects as the result of activity. I primarily consider activities such as scrubbing (i.e., moving a playhead back and forth on a timeline), watching, and re-watching. In place of the scrollbar, our canvas is an interactive timeline used to control video. Specifically, I consider how visualizing history of navigation activity can support exploration and analysis of video data.

I use a metaphor of "scratching-off" to reveal content that is at first hidden. Consider how the content of a video is hidden from the researcher upon first opening the video. The content exists as data stored on the disk, but is only made perceptible when the researcher skims or plays sections of the video. In a similar way, our timeline-based activity visualization uses frames from the video that are at first covered by an opaque

Figure 7.3: Video player with timeline showing activity-based visualization. Frames from segments of the video that have been viewed more are visible on the timeline, while unseen segments are occluded.

pattern. As the researcher views sections of the video, the corresponding sections of the pattern become transparent, revealing the frames underneath, as shown in Figure 7.3. The pattern appears to gradually get “scratched off" of the frames, similar to how one might scratch off an instant lottery ticket. This physical metaphor is reinforced by the possible interaction of dragging the cursor on the timeline to scan through the video, potentially moving back-and-forth to watch a segment multiple times.

For the sections that have been revealed, scrubbing the mouse over the timeline shows in a pop-up window a progression of frames for that period of video, providing access to some of the dynamics of the activity that has been previously watched and allowing a quick view of more detail than available in a static frame on the timeline.

In addition, this visualization can be used to represent activity related to any given segment of the video by adjusting the transparency of the activity layer in one of two ways: either as a function of the absolute time spent looking at a segment in relation to a user-specified threshold, or as a function of the percentage of time spent looking at one segment versus the whole video. The current implementation defines a point to transition from one strategy to the other, based on the total time spent with the video. When the total amount of time spent looking at the video is low, a representation of absolute time will likely be more useful, so that researchers can see what parts of the

Figure 7.4: Top: Design for visualizing temporally-linked activity oriented around related segments. Bottom: Visualization design for showing a segment overview from temporal boundaries.

video they have watched. Later, when more time has been spent watching the video, a representation showing relative time will likely be more useful, so that a researcher knows which parts of the video have been watched the most.

In addition to aggregating viewing activity, this visualization can be extended to represent relevant subsets of activity. For example, it may be helpful to represent which parts of a video were viewed during the last session, rather than the aggregate for the every session with the video. It may also be helpful to represent a chronology of activity over time, showing regions that have been viewed over multiple sessions in relation to the overall activity. This can be visualized by using multiple rows on the timeline, each depicting the pattern that results from different subsets of the activity history. It may also be useful to present information about the temporal nature of activity, such as a viewing pattern of jumping back and forth between two related segments. In this case, an overlay can be added consisting of ovals to identify the segments and lines linking the segments together (as shown in the top image of Figure 7.4), allowing rapid movement between segments.

A key aspect of this visualization technique is the ability to easily configure timeline visualizations to assist different stages of analysis. Another important envisioned use is to aid collaboration. Sharing activity histories may be a productive way to direct attention to specific areas that other analysts spent time viewing. Multiple collaborators' activity can be combined by using multiple timeline overlays with different colors or levels of transparencies. By looking across collaborators' activity, researchers can see areas of joint interest as well as "interesting" areas they may have not noticed.

7.3 Visualization Across Multiple Devices

A pervasive design element of ChronoViz is multiple linked visualizations, as discussed in Section 4.1. While much work has been done on multiple linked visualizations (see Section 3.2), and ChronoViz makes extensive use links between visualizations, an additional adaptation of this idea in ChronoViz is to allow linked visualization across multiple devices. An increasingly diverse collection of digital devices are commonly used in the daily course of work. For example, many people may be use multiple desktop or laptop computers, a tablet computer, and a smart phone on a daily basis. Allowing these devices to work together can enable new styles of interaction and overcome limitations present in interactions with a single device.

As part of my research with ChronoViz, I was interested in how this idea of an ecology of devices could support working with a single data set across multiple devices, and how an interactive system for visualization of data could effectively make use of a variable set of devices. Using multiple devices could not only expand how information could be presented, but also change the way in which researchers could interact with it.

Similar to the common idea in human-computer interaction research of designing systems to allow computers to do what they are good at, and human to do what they are good at, a key idea with multiple device visualization is to take advantage of the types of interaction for which each device is well suited. Each device that we interact with has properties that enable or promote some styles of interaction and prevent or discourage other styles of interaction. The modern desktop computer is a very flexible device, but some design elements create limitations. For example, the persistence of the mouse-and-keyboard for interacting with computers means that most interaction must be mediated by the things those devices allow, such as the use of a single cursor point for most interaction with on-screen elements. Another example is limited display space. Although multiple displays can be attached to a single computer, display space is still rather constrained and increasing one's own display space is a non-trivial matter.

One way around these limitations is to enable interaction through other devices. Consider digital pens and interactive paper. Pen and paper allows someone to spread information out over a large space. Since this technology is based on normal paper, display space is only limited by the amount of desk space or amount of paper. Paper can also be flexibly arranged, sorted, folded, and adapted. When interaction with digital notes takes place through paper instead of the onscreen representation, it gives researchers more flexibility for how the notes can be arranged and used. Depending on the style of notes and how they are being used to support analysis, the onscreen representation may not even need to be displayed. Different paper note sets can be stacked or grouped in ways that are difficult with the on-screen representation. In addition to extending display space and enabling flexible arrangement, use of the paper notes also supports collaboration in ways that may be difficult with an on-screen representation. Multiple note sets with multiple pens can be used at the same time to compare insights about a data set, or trade navigation control without needing to move seating arrangements so someone else is near the mouse.

Another alternative to the traditional desktop computer interfaces is the multi-touch tablet. These devices are particularly well-suited for fluid interactions involving zooming, panning, and manipulation. ChronoViz supports all of these types of interaction with the traditional interface, but these are not always fluid interactions. For example, with the map display of ChronoViz, the design privileges temporal navigation at the expanse of map manipulation. Clicking on the map will move the time point rather than pan or zoom the map, as is common with many interactive map displays. This design decision is based on the assumption that temporal navigation is a more frequent and important aspect of interacting with the map in ChronoViz. A multi-touch display can make this tradeoff less significant.

To explore this idea, I created an iPad app that functions as part of a ChronoViz device ecology. This app makes use of multi-touch interaction to enable easier zooming and movement in timelines and maps. Both types of visualizations behave similarly to their desktop-based counterparts for single-finger interaction; tapping and dragging moves the currently displayed time point. However, zooming is easily accomplished by using pinch gestures that have become standard in multitouch environments. The timescale of a timeline can be adjusted by pinching horizontally on the timeline, and the geographic scale of the map can be adjusting by pinching. In both cases, it is now possible to perform zooming actions without the need to change modes.

Chapter 8

Conclusion

8.1 Summary of Thesis Contributions

In this thesis I have presented research that aimed to answer the question of how tools to support analysis of multimodal temporal data should be designed. This was accomplished by gaining a better understanding of analysis activity, through direct observation of researchers while they engaged in the collection and analysis of multimodal data, and collection and analysis of interaction logs that recorded navigation of such data. I then applied these sources of insight to the design and implementation of new tools to aid researchers as they perform analysis.

The product of this research process is ChronoViz, a tool I created to support navigation and analysis of multimodal time-coded data. ChronoViz supports integration of a wide variety of data, including multiple video files, audio files, computer logs, sensor readings, paper notes, transcriptions, gaze data, motion capture data, and spatial trajectories. This integration is based on coordinated visualization of all of the data collected about a given activity sequence, allowing any one part of the data set to be used for navigation of the data set as a whole. Underlying the coordinated visualization are features for managing, aligning, and filtering data

ChronoViz represents several novel ideas for the design of interactive visualizations to support navigation and analysis of temporal data. Multiple linked visualizations that are independently configurable offer a high degree of flexibility for supporting analysis of different types of data and configuring the visualization environment to suit different research questions. Visualizations of different types of data, linked through time, allow one type of data to be used for navigation of another, and for the different data streams to provide context or elaboration of each other.

The design of ChronoViz was heavily based on this combination of extended interactions with researchers, participant observation with researchers collecting complex data, and logs of interactions. The collection and analysis of a large corpus of log file data from researchers as they performed analysis of video or other temporal data allowed me to have greater insight into the characteristics of temporal data navigation patterns exhibited during analysis. These characteristics in turn have helped to drive design as well as indicate possible ways that navigation could be better supported in the future.

ChronoViz itself is also a significant contribution as a tool for the research community. It is software that has been used by several research groups for actual research, and supports a wide range of usage styles. It has been and remains freely downloadable, and has been through several years of development. Although it is clearly "research software", it is quite stable and many researchers have found it to be useful even without taking advantage of many of the more advanced features.

Combined, the data and design that was described in this thesis improve the current state of understanding how to support analysis activity and provide an example of how this understanding can be applied. By instantiating these designs in software that was used for real analysis, I was able to observe their usage in practice and iteratively improve the designs based on these observations. While ChronoViz represents an advance in supporting this type of activity, the data collected about navigation also points to some clear future directions.

8.2 Future Directions

Although this thesis presents significant advancement in terms of understanding and supporting multimodal temporal data navigation, there are clearly remaining challenges and great potential for future work to improve on the work I reported here. One challenge of this research is that analysis activity is a moving target. Data analysis is an interaction between researchers, the data they collect, and the tools they use for analysis. The introduction of new technology, which can affect both the data that is collected and the tools that are available for analysis, changes the nature of the activity. ChronoViz, as new technology used for analysis, changes the nature of the analysis activity, but the insights provided by my observations and data still point toward directions for future work.

One major theme that is found among the remaining navigation challenges is better support for moving between different time scales. While ChronoViz provides some facility for this with temporal zooming and saved states, improvements can be made to both visualization and interaction techniques to better support navigation activities that occur at different scales. This may include multi-scale representations that tailor information to better support large- or small-scale navigation, and better support for quickly moving between time-scales.

A related challenge is providing better support for using one's own activity as a navigation aid. While I presented some work aimed at recording and presenting a viewing history, techniques for more intelligently processing the activity to enable navigation based on both recent and long-term history of activity could likely provide a good benefit to researchers. This could support better resumption of activity, by reflecting on activity the was recorded at the end of a previous session, or implicit annotation by considering where activity has been concentrated over many sessions. It could also support better navigation by automatically identifying regions of time that received attention and making it easy to return to those regions.

A second major theme is improved flexibility and clarity for annotation. One particular difficulty encountered by a number of researchers is taking advantage of the flexibility provided by the ChronoViz annotation system to best suit their specific analysis needs. While this thesis focused on navigation of data, giving more scrutiny to annotation and coding in future efforts could greatly benefit the overall effectiveness of tools to support analysis. In particular, this should include designing tools that support a range of analysis styles and evolution of annotation and categorization schemes. My observations reveal a clear need for better support for creating, applying, visualizing, and evolving categorization schemes.

A final area for future work is better automation of analysis activity. Although I don't have the data to quantize analysis efficiency, it seems as if advanced visualization capabilities in many cases may only keep balance with increased data collection capabilities, rather than lead to increases in analysis efficiency. To truly increase the speed of analysis, computational methods are also needed. While ChronoViz provides a framework for performing computational analysis, and early uses of this capability have indicated potential for automating some early stages of analysis. One particular area for future work is integration of computer vision techniques and other image processing techniques for both improved visualization, such as making motion more apparent, and automated annotation, such as identifying all frames where a particular person or object is present. Such techniques are powerful on their own, but integration with the interactive visualization and annotation features of ChronoViz would enable practical use for a variety of analysis tasks.

The research presented in this thesis provides the groundwork for further exploration of these ideas while at the same time providing powerful abilities that can improve analysis capabilities now. The combination of coordinated visualization of multiple streams of data with the informed design of annotation and analysis capabilities has made ChronoViz a powerful tool that has been applied to a wide range of research projects. It has enabled flexible analysis of rich collections of data streams, where before such analysis would have been unwieldy or impossible. While the design of ChronoViz has been demonstrated to be beneficial for many existing projects, the data and observations that I reported also indicate great potential for future efforts.

Appendix A

Plugin Design

A.1 Overview

The ChronoViz plugin structure supports two types of plugins: Plugins that are designed for data analysis, and plugins that are designed to expand visualization or data capabilities. In this appendix, I go into more detail on the design and implementation of these plugins, and provide an example of each type of plugin to illustrate how they work.

A.2 Analysis Plugins

Analysis plugins are written using the Python programming language, and take advantage of PyObjC[1], a project that provides a bi-directional bridge between code written in Python and Objective-C. Essentially, it makes it possible to use libraries and objects that are present in code written in one language by writing code in the other language. Objects and functions are dynamically translated at runtime, which makes it possible for scripts written in Python to be loaded in ChronoViz without any additional compilation or post-processing.

One side effect of using PyObjC is that it imposes a somewhat peculiar style on the Python code. To automatically translate Objective-C methods to Python methods, it

[1] http://pythonhosted.org/pyobjc/

replaces colons in Objective-C with underscores in Python. For example, an Objective-C method call that would be written `[annotation setTitle:@"The Title"]` gets translated to `annotation.setTitle_("The Title")` in Python.

A side effect of using PyObjC in combination with classes that are originally written in Objective-C is that it is impossible to test plugins outside of ChronoViz. Since plugins are designed to operate on ChronoViz data, this makes some sense, but it also makes debugging significantly more challenging. There are a number of Integrated Development Environments (IDEs), such as PyDev[2] and PyCharm[3], that offer sophisticated debugging tools while working on Python code, but these tools are inaccessible while running plugins from within ChronoViz. One way around this limitation that was developed by myself in collaboration with Ed Hutchins is to separate the analysis logic from the structural logic. An analysis logic class (e.g., containing the parts of code that actually perform analysis on data) can be written to operate on generic data classes without any specific ties to ChronoViz. A separate structural logic class can handle bi-directional translation between ChronoViz-specific data structures and specification of plugin parameters. In this way, the logic can be debugged using modern tools. Since the structural code is usually less complicated, this solves many development problems encountered with plugins. This design pattern currently needs to be implemented manually, but future work could provide automatic support for this separation by providing a pure Python API combined with dynamic generation of the the structural code that ties the Python code to the compiled Objective-C code.

The analysis plugin API works by writing plugins that are a subclass of a base plugin class. In the subclass, plugin authors needs to override two methods: a `setup` method that specifies parameters about the plugin, and a `performAnalysis` method that performs the actual analysis. The superclass provides support for defining the characteristics of the plugin, and creating and registering new annotations and data. By defining some characteristics within the setup method, it makes it possible to dynamically create a basic user interface for the specifying run-time parameters of a plugin. Figure A.1 shows an example example configuration window, defined by the example plugin in Section A.2.1. In this case, the user can select a time-series data set from the data sets

[2]http://pydev.org
[3]http://www.jetbrains.com/pycharm/

that are currently loaded in ChronoViz, and specify a parameter for the plugin to use during its analysis.

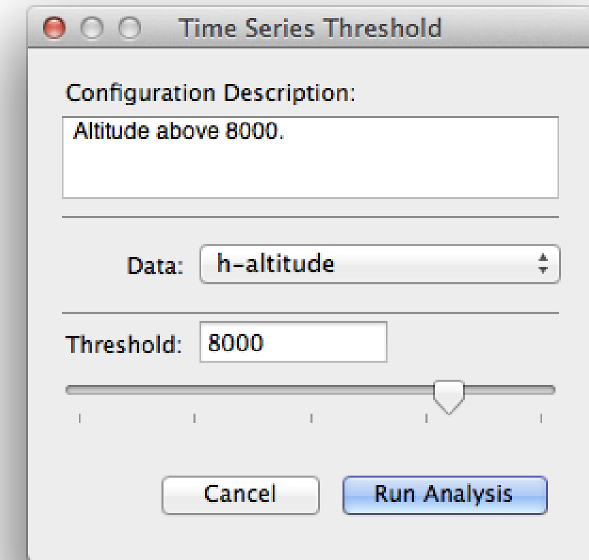


Figure A.1: Plugin configuration window that allows the user to specify a data set and parameters before running the plugin.

A.2.1 Example Analysis Plugin

This example is a plugin meant to demonstrate the basic elements of the Chrono-Viz Data Analysis Plugin API. The logic of the plugin is very simple: it simply creates annotations that represent portions of a time-series data set that are above a user-defined threshold. However, the structural code demonstrates several features available in the Analysis Plugin API.

In the `setup` method, various characteristics are set that define the appearance and behavior of the plugin within ChronoViz. The plugin is given a “display name” that defines how the plugin will appear in menus and windows (line 20). Then, the plugin specifies the type and quantity of data on which it performs analysis (lines 23 - 24). In this case, a single time-series data set named "Data" is specified. While the name is uninformative in this case (since only a single data set is required), in more complex plugins the name can be used to differentiate data sources (e.g., "Group Leader Notes" versus "Participant A Notes") or data meaning (e.g., "Altitude" versus "Airspeed"). Finally, the plugin specifies a parameter that can be set by the user when the plugin is run (lines 26 - 29). Since this plugin computes a threshold, the parameter determines the threshold level.

In the `performAnalysis` method, the plugin first retrieves the threshold parameter and the data points that were specified when the user ran the plugin (lines 33 - 34). Next, the plugin defines an annotation category that will be assigned to all of the annotations that are created during the analysis (lines 37 - 39). Finally, the plugin finds the threshold regions by looping over all of the data points, and creating annotations that correspond to those regions (lines 43 - 60).

```
1  #
2  #   Threshold.py
3  #   Threshold Plugin for ChronoViz
4  #
5
6  from Foundation import *
7  import objc
8
9  PluginClass = objc.lookUpClass('AnnotationDataAnalysisPlugin')
10 Annotation = objc.lookUpClass('Annotation')
11 TimeCodedDataPoint = objc.lookUpClass('TimeCodedDataPoint')
12 PluginParameter = objc.lookUpClass('PluginParameter')
13
14 class Threshold(PluginClass):
15
16     inThreshold = False
17
18     def setup(self):
19          # Sets the name of the plugin in the menu
20         self.setDisplayName_("Time Series Threshold")
21
22         # Determines which data sets the user can choose
```

```
        self.setDataVariableClass_("TimeSeriesData")
        self.dataVariable = self.addDataVariable_("Data")

        self.thresholdParameter = self.addInputParameter_("Threshold")
        self.thresholdParameter.setParameterValue_(20)
        self.thresholdParameter.setMinValue_(1)
        self.thresholdParameter.setMaxValue_(10000)

    def performAnalysis(self):
        # Retrieve values set by user at runtime
        theThreshold = self.thresholdParameter.parameterValue()
        dataPoints = self.dataVariable.dataPoints()

        # Set up an Annotation Category to give to annotations
        dataSetName = self.dataVariable.dataSet().name()
        ResultsCat = self.categoryForName_(dataSetName + " above " + str(theThreshold))
        ResultsCat.autoColor()
        annotation = 0

        # Compute threshold by looping over data points
        for data in dataPoints:
            dataValue = data.numericValue()
            if (dataValue > theThreshold):
                if self.inThreshold:
                    pass
                else:
                    self.inThreshold = True
                    annotation = self.newAnnotationAtTime_(data.time())
                    annotation.setTitle_("Above Threshold")
                    annotation.setIsDuration_(True)
                    annotation.addCategory_(ResultsCat)
            else:
                if self.inThreshold:
                    self.inThreshold = False
                    annotation.setEndTime_(data.time())
```

```

        if self.inThreshold:
            annotation.setEndTime_(data.time())
```

A.3 Visualization and Data Plugins

Visualization and Data Plugins create new types of visualizations or enable ChronoViz to load new types of data. The plugins are written primarily using Objective-C, and rely on an "App Proxy" object that provides links to various ChronoViz functions. There are two main ways that functionality can be added: creating a view manually and handling connections within the plugin, or registering a data class and view class with ChronoViz so that the view will automatically be created and linked when the data is imported.

One design pattern that is made possible through these plugins is to use existing web-based visualizations with a JavaScript wrapper to link them in to ChronoViz. The examples in this section are from a plugin I wrote to link the Discursis (Angus et al., 2012) conversation analysis visualization to data in ChronoViz. There are three components to this plugin that are included below. First is a plugin class that creates the view for the visualization and connects it to ChronoViz. Second is a view class that provides ChronoViz with the necessary connections for time linkage and provides a web view to show the existing web-based discourses plugin. Third is a set of JavaScript functions that create a bridge between the JavaScript code and the Objective-C code.

A.3.1 Example Plugin Code

In the following code sample is a selection of methods from the main plugin class that gets loaded by ChronoViz. The class has an initialization method that gets called to perform any setup that is required when ChronoViz starts. In this case, it creates a menu item to open up the Discursis window. Following the initialization method are two methods to retrieve and set the current time in ChronoViz, and finally there is a method to create the Discursis view and let ChronoViz know that it should be linked in time.

```
//
//  ChronoVizDiscursisPlugin.m
//  ChronoVizDiscursisPlugin
//

#import "ChronoVizDiscursisPlugin.h"
#import "DPAppProxy.h"
#import "ChronoVizDiscursisWindowController.h"
#import "VideoProperties.h"
#import "DiscursisView.h"

@implementation ChronoVizDiscursisPlugin

- (id) initWithAppProxy:(DPAppProxy *)appProxy
{
        self = [super init];
        if (self != nil) {

        app = appProxy;

        NSMenuItem *menuItem =
            [[NSMenuItem alloc]
                initWithTitle:@"Discursis"
                       action:@selector(openDiscursisWindow:)
                keyEquivalent:@""];
        [menuItem setTarget:self];
        [appProxy addMenuItem:menuItem toMenuNamed:@"File"];
        [menuItem release];

        }
        return self;
}

- (QTTime)currentTime
{
    return [app currentTime];
}

```

```
- (void)setCurrentTime:(QTTime)time fromSender:(id)sender
{
    [app setCurrentTime:time fromSender:sender];
}

- (void) openDiscursisWindow:(id)sender
{
    if(!vizWindow)
    {
        vizWindow =
            [[ChronoVizDiscursisWindowController alloc] init];
        [[vizWindow window] makeKeyAndOrderFront:self];

        [[vizWindow discursisView] reloadData];

        [app addAnnotationView:[vizWindow discursisView]];
    }
    else
    {
        [[vizWindow window] makeKeyAndOrderFront:self];
    }

}



@end
```

A.3.2 Example View Code

This is a selection of methods from the "view" class that creates a web view to show the Discursis visualization and translate actions in from the visualization to time commands within ChronoViz. This class works by creating an associative array of transcript indices to times. This array is created when data is loaded, then called upon when the visualized transcript elements are clicked in the visualization.

```
//
//  DiscursisView.m
//  ChronoViz
//

#import "DiscursisView.h"
#import "DataSource.h"
#import "ChronoVizDiscursisPlugin.h"
#import "NSStringTimeCodes.h"

@implementation DiscursisView


-(void)setData:(TranscriptData*)source
{
        [source retain];
        [data release];
        data = source;

        [self reloadData];
}

-(void)handleTimeClick:(NSTimeInterval)time
{
        currentTime = QTMakeTimeWithTimeInterval(time);

    [[ChronoVizDiscursisPlugin defaultPlugin]
        setCurrentTime:currentTime
            fromSender:self];
}
```

```
-(void)handleIndexClick:(NSInteger)transcriptIndex
{
    NSLog(@"Index Clicked %i",(int)transcriptIndex);

    if(!indicesToTimes || ([indicesToTimes count] <=
        transcriptIndex))
    {
        currentTime = QTMakeTimeWithTimeInterval(((CGFloat)
            transcriptIndex) * 0.1);
        [[ChronoVizDiscursisPlugin defaultPlugin]
            setCurrentTime:currentTime fromSender:self];
    }
    else
    {
        id object = [indicesToTimes objectAtIndex:
            transcriptIndex];
        if(object != [NSNull null])
        {
            currentTime = QTMakeTimeWithTimeInterval([object
                doubleValue]);
            [[ChronoVizDiscursisPlugin defaultPlugin]
                setCurrentTime:currentTime fromSender:self];
        }
    }

}


- (NSTimeInterval)timeFromMetadata:(NSString*)metadata
{
    NSArray *components =
        [metadata componentsSeparatedByString:@","];
    for(NSString *component in components)
    {
        NSRange startRange =
            [component rangeOfString:@"StartTime"
                                options:NSCaseInsensitiveSearch];
```

```
        if(startRange.location != NSNotFound)
        {
            return [[[component componentsSeparatedByString:@"
               "] lastObject] timeInterval];
        }
    }

    return -1;
}

-(int)loadedPlotId:(NSString*)plotID
{
    if([[loadedPlots allKeys] containsObject:plotID])
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

-(void)loadPlot:(WebScriptObject*)plot
         withId:(NSString*)plotID
{
    [loadedPlots setObject:plot forKey:plotID];
}

-(void)registerIndex:(int)index
        withMetadata:(NSString*)metadata
{
    NSTimeInterval time = [self timeFromMetadata:metadata];
    NSNumber *timeNumber = [NSNumber numberWithDouble:time];

    int diff = index - [indicesToTimes count];

    while(diff > 0)
    {
```

```
        diff--;
        [indicesToTimes addObject:[NSNull null]];
    }

    if([indicesToTimes count] == index)
    {
        [indicesToTimes addObject:timeNumber];
    }
    else
    {
        [indicesToTimes replaceObjectAtIndex:index withObject:timeNumber];
    }

}

-(void)addDiscursisProject:(NSString*)projectString
{
    NSCharacterSet *quotes = [NSCharacterSet characterSetWithCharactersInString:@"'\""];
    NSRange start = [projectString rangeOfString:@"("];
    NSRange end = [projectString rangeOfString:@")"];
    if(start.location != NSNotFound)
    {
        NSArray *components = [[[projectString substringToIndex:end.location] substringFromIndex:(start.location + 1)] componentsSeparatedByString:@","];
        NSString *projectID = [[components objectAtIndex:0] stringByTrimmingCharactersInSet:quotes];
        NSString *projectName = [[components objectAtIndex:1] stringByTrimmingCharactersInSet:quotes];

        [projects setObject:projectID forKey:projectName];

        NSLog(@"Register Project: %@ %@",projectID,projectName);
    }
}
```

```
-(void)loadFinished:(id)sender
{
    dispatch_async(dispatch_get_main_queue(), ^{
        if([projectSelectionWindow isVisible])
        {
            [NSApp endSheet:projectSelectionWindow];
            [projectSelectionWindow orderOut:self];
        }

        [self selectDiscursisProject];

    });
}

-(void)loadProjectNamed:(NSString*)projectName
{
    NSString *projectID = [projects objectForKey:projectName];

    if(!projectID)
        return;

    NSString *command = [NSString stringWithFormat:@"
       showProject('%@','%@');",projectID,projectName];

    [[webView windowScriptObject] evaluateWebScript:command];

    NSRect currentFrame = [self frame];
    NSRect diddleFrame = currentFrame;
    diddleFrame.size.width = currentFrame.size.width + 1;
    [self setFrame:diddleFrame];
    [self setFrame:currentFrame];
}


@end
```

A.3.3 Example Javascript Code

The following set of JavaScript functions is added to the existing Discursis visualization. The first set of functions translate JavaScript functions to Objective-C method calls. The second set of functions override existing parts of the Discursis code base to add calls to the translation functions in the first set when elements are clicked and when data is loaded.

```
/* These functions send messages to ChronoViz */

function chronovizDoTimeClick(time) {
    if ( chronoVizView ) {
        chronoVizView.handleTimeClick_(time);
    }
}

function chronovizDoIndexClick(transIndex) {
    if ( chronoVizView ) {
        chronoVizView.handleIndexClick_(transIndex);
    }
}

function chronovizDoItemSelect(item) {
    if ( chronoVizView ) {
        chronoVizView.handleItemWithIndex_andMetadata_(item.
            index,item.metadata);
    }
}

function chronovizAddProject(theString) {
    if ( chronoVizView ) {
        chronoVizView.addDiscursisProject_(theString);
    }
}

function chronovizLoadFinished(theString) {
    if ( chronoVizView ) {
        chronoVizView.loadFinished_(theString);
```

```
    }
}

function chronovizLoadPlotItems(thePlot) {
    if ( chronoVizView ) {

        var firstItem = thePlot.getFirstItem();
        var lastItem = thePlot.getLastItem();
        for(var index = firstItem.getIndex(); index < lastItem.
            getIndex(); index++)
        {
            var item = thePlot.getItem(index);
            chronoVizView.registerIndex_withMetadata_(index,
                crps.data.utterances.fmtMetadata(item.metadata))
                ;
        }

        chronoVizView.loadPlot_withId_(thePlot,thePlot.getId())
            ;
    }
}

/* These functions override functions in the standard Discursis
    code */

crps.uic.plot.PlotItem.prototype.select = function(event) {
        if (!this.plot.allowSelect()) {
            return;
        }
        event.cancelBubble = true;
        if (this.selected) {
            return;
        }
        chronovizDoIndexClick(this.index);
        this.selected = true;
        this._border.show();
        this.fireEvent('select', this);
```

```
        this.plot.highlight(this.index, this.indexB || this.
            index);
}

crps.uic.plot.Plot.prototype.originalDrawPlot = crps.uic.plot.
    Plot.prototype.drawPlot;

crps.uic.plot.Plot.prototype.drawPlot = function (plotEl,
    drawOptions) {
    var returnVal = this.originalDrawPlot(plotEl, drawOptions);
     chronovizLoadPlotItems(this);
     return returnVal;
};
```

References

Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., … Cohen, M. (2004). Interactive digital photomontage. *ACM Transactions on Graphics (TOG)*, *23*(3), 294–302.

Aigner, W., Miksch, S., Schumann, H., & Tominski, C. (2011). *Visualization of time-oriented data*. Springer.

André, P., Wilson, M. L., Russell, A., Smith, D. A., Owens, A., et al. (2007). Continuum: designing timelines for hierarchies, relationships and scale. In *Proceedings of the 20th annual ACM symposium on user interface software and technology* (pp. 101–110).

Angus, D., Watson, B., Smith, A., Gallois, C., & Wiles, J. (2012). Visualising conversation structure across time: insights into effective doctor-patient consultations. *PLoS One*, *7*(6), e38014. doi: 10.1371/journal.pone.0038014

Bade, R., Schlechtweg, S., & Miksch, S. (2004). Connecting time-oriented data and information to a coherent interactive visualization. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 105–112).

Barnes, C., Goldman, D. B., Shechtman, E., & Finkelstein, A. (2010). Video tapestries with continuous temporal zoom. *ACM Transactions on Graphics (TOG)*, *29*(4), 89.

Becker, R., & Cleveland, W. (1987). Brushing scatterplots. *Technometrics*.

Bederson, B., & Hollan, J. (1994). Pad++: a zooming graphical interface for exploring alternate interface physics. *Proceedings of the 7th annual ACM symposium on User Interface Software and Technology*.

Bisantz, A., Marsiglio, S., & Munch, J. (2005). Displaying uncertainty: Investigating the effects of display format and specificity. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *47*(4), 777–796.

Bouvier-Zappa, S., Ostromoukhov, V., & Poulin, P. (2007). Motion cues for illustration of skeletal motion capture data. In *Proceedings of the 5th international symposium* *on non-photorealistic animation and rendering* (pp. 133–140).

Card, S., Mackinlay, J., & Shneiderman, B. (1999). *Readings in information visualization: Using vision to think.* Morgan Kaufman.

Cleveland, W., McGill, M., & McGill, R. (1988). The shape parameter of a two-variable graph. *Journal of the American Statistical Association*, *83*(402), 289 - 300.

Cleveland, W., & McGill, R. (1987). Graphical perception: The visual decoding of quantitative information on graphical displays of data. *Journal of the Royal Statistical Society. Series A (General)*, *150*(3), 192-229.

Correa, C. D., & Ma, K.-L. (2010). Dynamic video narratives. *ACM Transactions on Graphics (TOG)*, *29*(4), 88.

Davis, M. (1993). Media streams: An iconic visual language for video annotation. In *Proceedings 1993 IEEE symposium on visual languages* (pp. 196–202).

Davis, M. (2000). Media streams: an iconic visual language for video representation. *Readings in Human-Computer Interaction: Toward the Year 2000*, 854–866.

Eckmann, J., Kamphorst, S., & Ruelle, D. (1987). Recurrence plots of dynamical systems. *EPL (Europhysics Letters)*, *4*, 973.

Furnas, G. (1986, April). Generalized fisheye views. In *CHI '86: Proceedings of the SIGCHI conference on human factors in computing systems* (p. 16 - 23). New York: ACM.

Goldman, D., Curless, B., Salesin, D., & Seitz, S. (2006, Jan). Schematic storyboarding for video visualization and editing. *International Conference on Computer Graphics and Interactive Techniques*.

Goodwin, C. (1994). Professional vision. *American anthropologist*, *96*(3), 606–633.

Hagedorn, J., Hailpern, J., & Karahalios, K. (2008). VCode and VData: Illustrating a new framework for supporting the video annotation workflow. In *Extended abstracts of AVI 2008.*

Hanson, C., & Hirst, W. (1989). On the representation of events: A study of orientation, recall, and recognition. *Journal Experimental Psychology General*, *118*(2), 136-147.

Heer, J., & Agrawala, M. (2006). Multi-scale banking to 45 degrees. *IEEE Transactions on Visualization and Computer Graphics*, *12*(5), 701 - 708.

Heer, J., Kong, N., & Agrawala, M. (2009, Apr). Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations. In

CHI '09: Proceedings of the SIGCHI conference on human factors in computing systems.

Hill, W. C., Hollan, J. D., Wroblewski, D., & McCandless, T. (1992). Edit wear and read wear. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 3–9).

Hilpoltsteiner, M. (2006). *Recreating movement (http://www.recreating-movement.com).* Retrieved October 2011, from http://www.recreating-movement.com/

Hodges, S., Williams, L., Berry, E., Izadi, S., Srinivasan, J., Butler, A., … Wood, K. (2006). Sensecam: A retrospective memory aid. *UbiComp 2006: Ubiquitous Computing*, 177–193.

Hofmann, C., Hollender, N., & Fellner, D. (2009). Task-and process-related design of video annotation systems. In *Proc. mensch & computer '09.*

Hollan, J. D., Hutchins, E., & Kirsh, D. (2000). Distributed cognition: toward a new foundation for human-computer interaction research. *ACM transactions on computer-human interaction*, *7*(2), 174–196.

House, D., Bair, A., & Ware, C. (2006, Jul). An approach to the perceptual optimization of complex visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, *12*(4), 509 – 521. doi: 10.1109/TVCG.2006.58

Hutchins, E. (1995). *Cognition in the wild.* The MIT Press.

Inselberg, A., & Dimsdale, B. (1990). Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the 1st conference on visualization'90* (pp. 361–378).

Kang, H.-W., Chen, X.-Q., Matsushita, Y., & Tang, X. (2006). Space-time video montage. In *2006 IEEE computer society conference on computer vision and pattern recognition* (Vol. 2, pp. 1331–1338).

Kipp, M. (2008). Spatiotemporal coding in ANVIL. In *6th international conference on language resources and evaluation (lrec-08).*

Krasner, G. E., Pope, S. T., et al. (1988). A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, *1*(3), 26–49.

Lamping, J., Rao, R., & Pirolli, P. (1995). A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95: Proceedings of the SIGCHI conference on human factors in computing systems.*

Li, Y., Zhang, T., & Tretter, D. (2001). An overview of video abstraction techniques. *Hewlett-Packard Labs Technical Report*.

Lu, S., Harter, D., & Graesser, A. (2009). An empirical and computational investigation of perceiving and remembering event temporal relations. *Cognitive Science*, *33*(3), 345–373.

Mackay, W., & Beaudouin-Lafon, M. (1998). DIVA: exploratory data analysis with multimedia streams. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 416–423).

McCloud, S. (1994). *Understanding comics*. Harper Paperbacks.

Muller, W., & Schumann, H. (2003). Visualization methods for time-dependent data-an overview. In *Proceedings of the 2003 winter simulation conference* (Vol. 1, pp. 737–745).

Newtson, D. (1973). Attribution and the unit of perception of ongoing behavior. *Journal of Personality and Social Psychology*, *28*(1), 28.

North, C., & Shneiderman, B. (2000). Snap-together visualization: a user interface for coordinating visualizations via relational schemata. *Proceedings of the working conference on Advanced visual interfaces*.

Nunes, M., Greenberg, S., Carpendale, S., & Gutwin, C. (2006). Timeline: Video traces for awareness. *Video Proceedings of ACM Computer-Supported Collaborative Work*.

Pea, R., & Hoffert, E. (2007). Video workflow in the learning sciences: Prospects of emerging technologies for augmenting work practices. In R. Goldman, R. Pea, B. Barron, & S. Derry (Eds.), *Video research in the learning sciences* (p. 427-460).

Pea, R., & Lemke, J. (2007). Sharing and reporting video work. In S. Derry (Ed.), *Guidelines for video research in education: Recommendations from an expert panel.* Data Research and Development Center.

Pea, R., Mills, M., Rosen, J., Dauber, K., Effelsberg, W., & Hoffert, E. (2004). The DIVER project: Interactive digital video repurposing. *IEEE MultiMedia*, *11*(1).

Plaisant, C. (2004). The challenge of information visualization evaluation. In *Proceedings of the working conference on advanced visual interfaces* (pp. 109–116).

Plaisant, C., Milash, B., Rose, A., Widoff, S., & Shneiderman, B. (1996). Lifelines: visualizing personal histories. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 221–227).

Plumlee, M., & Ware, C. (2003). An evaluation of methods for linking 3D views. In *Proceedings of the 2003 symposium on interactive 3D graphics.*

Plumlee, M., & Ware, C. (2006). Zooming versus multiple window interfaces: Cognitive costs of visual comparisons. *ACM Transactions on Computer-Human Interaction*, *12*(2), 179–209.

Pongnumkul, S., Wang, J., Ramos, G., & Cohen, M. (2010). Content-aware dynamic timeline for video browsing. In *Proceedings of the 23nd annual ACM symposium on user interface software and technology* (pp. 139–142).

Pope, A., Kumar, R., Sawhney, H., & Wan, C. (1998). Video abstraction: summarizing video content for retrieval andvisualization. *Signals*.

Rao, R., & Card, S. (1994). The table lens: merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*.

Richter, H. A., Brotherton, J. A., Abowd, G. D., & Truong, K. N. (1999). A multi-scale timeline slider for stream visualization and control.

Roberts, J. (2007, July). State of the art: Coordinated & multiple views in exploratory visualization. In *Fifth international conference on coordinated and multiple views in exploratory visualization* (p. 61 - 71). Zurich.

Romero, M., Vialard, A., Peponis, J., Stasko, J., & Abowd, G. (2011). Evaluating video visualizations of human behavior. In *Proceedings of the 2011 annual conference on human factors in computing systems* (pp. 1441–1450).

Roschelle, J., & Goldman, S. (1991). VideoNoter: A productivity tool for video data analysis. *Behavior Research Methods*, *23*(2), 219–224.

Sanderson, P., & Fisher, C. (1994). Exploratory sequential data analysis: foundations. *Human-Computer Interaction*, *9*(3), 251-317.

Sanderson, P., Scott, J., Johnston, T., Mainzer, J., Watanabe, L., & James, J. (1994). Macshapa and the enterprise of exploratory sequential data analysis (esda). *International Journal of Human-Computer Studies*, *41*(5), 633-681.

Schank, R., & Abelson, R. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures.* Hillsdale, NJ: Lawrence Erlbaum.

Schwan, S., & Garsoffky, B. (2004). The cognitive representation of filmic event summaries. *Applied Cognitive Psychology*, *18*, 37-55.

Shneiderman, B. (1994). Dynamic queries for visual information seeking. *Software, IEEE*, *11*(6), 70–77.

Shneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of IEEE symposium on visual languages* (pp. 336–343).

Silva, S., Madeira, J., & Santos, B. S. (2007). There is more to color scales than meets the eye: a review on the use of color in visualization. In *IV'07. 11th international conference on information visualization* (pp. 943–950).

Smith, M. A., & Kanade, T. (1998). Video skimming and characterization through the combination of image and language understanding. In *Proceedings of 1998 IEEE international workshop on content-based access of image and video database* (pp. 61–70).

Speer, N., Swallow, K., & Zacks, J. (2003). Activation of human motion processing areas during event perception. *Cognitive, Affective, and Behavioral Neuroscience*, *3*(4), 335-345.

Suchman, L. A. (1987). *Plans and situated actions: the problem of human-machine communication*. Cambridge university press.

Swayne, D., Buja, A., & Lang, D. (2003). Exploratory visual analysis of graphs in ggobi. In *Proceedings of the 3rd international workshop on distributed statistical computing.*

Truong, B. T., & Venkatesh, S. (2007). Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, *3*(1), 3.

Tukey, J. (1977). Exploratory data analysis. *Addison-Wesley Series in Behavioral Science: Quantitative Methods, Reading, Mass*, *1*.

Wang, T., Mei, T., Hua, X.-S., Liu, X.-L., & Zhou, H.-Q. (2007). Video collage: A novel presentation of video sequence. In *2007 ieee international conference on multimedia* (pp. 1479–1482).

Wang, T. D., Plaisant, C., Quinn, A. J., Stanchak, R., Murphy, S., & Shneiderman, B. (2008). Aligning temporal data by sentinel events: discovering patterns in electronic health records. In *Proceedings of the twenty-sixth annual SIGCHI conference on human factors in computing systems* (pp. 457–466).

Wang Baldonado, M. Q., Woodruff, A., & Kuchinsky, A. (2000). Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on advanced visual interfaces* (pp. 110–119).

Ware, C. (1988). Color sequences for univariate maps: Theory, experiments and principles. *Computer Graphics and Applications, IEEE*, *8*(5), 41–49.

Ware, C. (2004). *Information visualization: Perception for design*. San Francisco: Morgan Kaufman.

Weaver, C. (2004). Building highly-coordinated visualizations in improvise. In *INFOVIS 2004: Ieee symposium on information visualization* (pp. 159–166).

Weibel, N., Fouse, A., Emmenegger, C., Friedman, W., Hutchins, E., & Hollan, J. (2012). Digital pen and paper practices in observational research. In *CHI '12: Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1331–1340).

Weston, C., Gandell, T., Beauchamp, J., McAlpine, L., Wiseman, C., & Beauchamp, C. (2001). Analyzing interview data: The development and evolution of a coding system. *Qualitative Sociology*, *24*(3), 381–400.

Wittenburg, P., Brugman, H., Russel, A., Klassmann, A., & Sloetjes, H. (2006). ELAN: A professional framework for multimodality research. In *Proceedings of LREC 2006.*

Wongsuphasawat, K., Guerra Gómez, J. A., Plaisant, C., Wang, T. D., Taieb-Maimon, M., & Shneiderman, B. (2011). Lifeflow: visualizing an overview of event sequences. In *Proceedings of the 2011 annual conference on human factors in computing systems* (pp. 1747–1756).

Woods, D., & Fassnacht, C. (2007). *Transana v2.2x.* Retrieved from http://www.transana.org

Xiao, R., Wang, Y., Pan, H., & Wu, F. (2008, Apr). Automatic video summarization by spatio-temporal analysis and non-trivial repeating pattern detection. *Image and Signal Processing, 2008. CISP '08. Congress on*, *4*, 555 – 559. doi: 10.1109/CISP.2008.601

Yeung, M., Yeo, B.-L., & Liu;, B. (1996, May). Extracting story units from long programs for video browsing and navigation. *Proceedings of the Third IEEE International Conference on Multimedia Computing and Systems*, 296 – 305. doi: 10.1109/MMCS.1996.534991

Zacks, J. M. (2004, Jan). Using movement and intentions to understand simple events. *Cognitive Science*.

Zacks, J. M., Speer, N. K., Vettel, J. M., & Jacoby, L. L. (2006). Event understanding and memory in healthy aging and dementia of the alzheimer type. *Psychology and aging*, *21*(3), 466–82. doi: 10.1037/0882-7974.21.3.466

Zacks, J. M., & Tversky, B. (2001). Event structure in perception and conception. *Psychological Bulletin*, *127*(1), 3-21.

Zacks, J. M., Tversky, B., & Iyer, G. (2001). Perceiving, remembering, and communicating structure in events. *Journal of Experimental Psychology General*, *130*(1), 29–58.

Zimmerman, P. H., Bolhuis, J. E., Willemsen, A., Meyer, E. S., & Noldus, L. P. J. J. (2009). The Observer XT: A tool for the integration of and synchronization of multimodal signals. *Behavioral Research Methods*, *41*(3).