VCode and VData: Illustrating a new Framework for Supporting the Video Annotation Workflow

Joey Hagedorn, Joshua Hailpern, Karrie G. Karahalios
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana, IL 61801 USA
{hagedorn, jhailpe2, kkarahal}@uiuc.edu

ABSTRACT

Digital tools for annotation of video have the promise to provide immense value to researchers in disciplines ranging from psychology to ethnography to computer science. With traditional methods for annotation being cumbersome, time-consuming, and frustrating, technological solutions are situated to aid in video annotation by increasing reliability, repeatability, and workflow optimizations. Three notable limitations of existing video annotation tools are lack of support for the annotation workflow, poor representation of data on a timeline, and poor interaction techniques with video, data, and annotations. This paper details a set of design requirements intended to enhance video annotation. Our framework is grounded in existing literature, interviews with experienced coders, and ongoing discussions with researchers in multiple disciplines. Our model is demonstrated in a new system called VCode and VData. The benefit of our system is that is directly addresses the workflow and needs of both researchers and video coders.

Categories and Subject Descriptors

H.5.1 [**Multimedia Information Systems**]: Evaluation/methodology Video
; H.5.2 [**User Interfaces**]: Graphical user interfaces

General Terms

Design, Human Factors, Measurement

Keywords

Graphical user interfaces (GUI), Annotation, Video

1. INTRODUCTION

Human behavior does not naturally lend itself to being quantifiable. Yet time and again, researchers in disciplines ranging from psychology to ethnography to computer science, are forced to analyze as if it was quantified. Those in human centered domains can now rely on video annotation to provide them with measures on which to draw conclusions. Unlike transcription, which is akin to what a court stenographer does, annotation is the marking of movements, sounds, and other such events (with or without additional metadata such as rankings). The emergence of technology as a tool to aid in video annotation has raised the possibility of increasing reliability, repeatability, and workflow optimizations [6]. Three notable limitations of existing video annotation tools are lack of support for the annotation workflow, poor representation of data on a timeline, and poor interaction techniques with video, data, and annotations. This paper details a set of requirements to guide the design of video annotation tools. Our model is the direct result of an analysis of existing tools, current practices by researchers, and workflow difficulties experienced by real-world video coders. By understanding what data researchers are looking to gather, and the shortcomings of existing techniques and technology utilized by coders, we believe that we have created a framework for video annotation that can reach across disciplines. Our model is demonstrated through the design and construction of our new system VCode and VData (Figure 1); two fully functional, open-source tools which bridge the video annotation workflow.

The primary contribution of this paper is the set of design requirements for facilitating a system conducive to video annotation. Specifically, we demonstrate how a system could be designed and built to meet these requirements through a

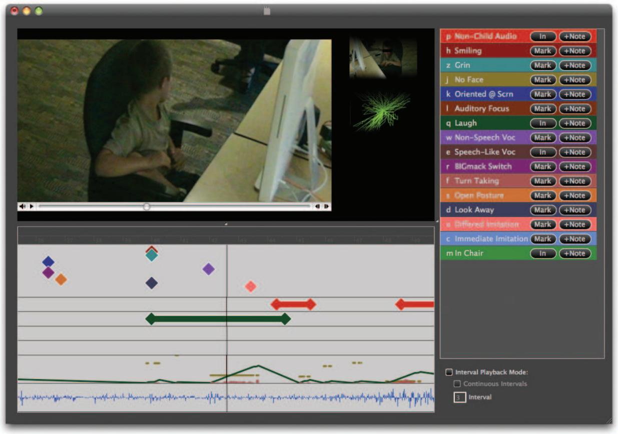

p Non-Child Audio In +Note
h Smiling Mark +Note
z Grin Mark +Note
j No Face Mark +Note
k Oriented @ Scrn Mark +Note
l Auditory Focus Mark +Note
q Laugh In +Note
w Non-Speech Voc Mark +Note
e Speech-Like Voc In +Note
r BIGmack Switch Mark +Note
f Turn Taking Mark +Note
s Open Posture Mark +Note
d Look Away Mark +Note
x Differed Imitation Mark +Note
c Immediate Imitation Mark +Note
m In Chair In +Note
Interval Playback Mode:
Continuous Intervals
3 Interval

Figure 1: The VCode application graphically represents the behaviors being coded as marks on a timeline. It is easy to see correlation between the marks on the timeline and sensor data displayed below.

AVI '08 May 28-30, 2008, Naples, Italy

set of carefully designed interfaces and graphical representations of data.

2. RELATED WORK

2.1 Video Coding In Practice

The analysis of human behavior is a study that dates back hundreds of years. This has ranged from anthropological ethnographies [10] to psychological evaluations. As technology has developed, the use of video and creation of annotation techniques have aided researchers by providing a referable document that can be used as evidence to back up claims and observations made [21, 16, 20]. These techniques involve detailed event logging on paper, specifying features such as durations, ratings/levels, and time-stamps [15].

To ensure a reliable set of data from annotation, researchers perform agreement calculations between coders [7]. This agreement is utilized throughout the data gathering process (by testing some small percentage of data segments to ensure consistency throughout), but also during training of coders (to decide when they fully understand what events they are looking for). There are many techniques for calculating agreement including Cohens Kappa [11], Cochran's Q-test, and Point-By-Point Agreement. Regardless, the management of data with traditional means is considered "cumbersome" [20].

2.2 Video Coding Tools

Digital annotation tools have demonstrated significant benefits from simple copy/paste and undo to increased quality of coding by the facilitation of multiple passes on video and graphical representations [4, 6]. A timeline is commonly utilized in these tools and is familiar without extensive training [1]. Existing research also indicates that presenting coders with secondary or sensor data on a timeline helps them outperform coders without sensor data [6]. Increased accuracy, quality, and speed not only enhance the data collected, but also allow for more annotation to be conducted in the same amount of time. In addition to the computational benefits of digital annotation tools, they also provide a controllable mechanism for different forms of reliable video playback. [4].

One critical limitation of existing tools is poor representation of data on a timeline and utilization of screen real-estate. For example, the VACA solution, while utilizing minimal screen real-estate by condensing all annotations to one large easy to read track, presents a problem with overlapping and simultaneous events [6]. The VisSTA solution takes the contrary approach by showing many small vertically tiled tracks. Though this allows for a good comparative view, reading individual annotations & holistic interpretation is difficult due to scrolling [4]. These and other existing solutions have not successfully dealt with this problem [14, 3, 13, 12, 1, 2, 22, 18].

Another limitation of current annotation tools is poor interaction techniques with video and data. Though robust functionality is provided for playback, controls can be cumbersome & overly complex, (e.g. [4]). Too many windows resulting in an over-saturation of information, imprecise video interaction & annotation or rigid, inSSexible marking interfaces (e.g. [4, 14, 12, 18, 3]). Each of these are common stumbling blocks which could result in unreliable data.

One last limitation is lack of support for the full annotation workflow; 1) collect video 2) create segments to code 3) train coders/demonstrate reliability 4) gather data 5) perform regular checks on reliability & discuss discrepancies 6) perform data analysis. Many tools support small portions of this workflow (i.e. simply facilitating segmentation, annotation, or reliability [22, 6, 12]), but with each break in the process researchers can become delayed. Without export/import data reentry is required. Technology is situated to optimize this process.

Researchers have also explored dialogue transcription [17, 13, 12], tagging [2, 12], scene based automatic annotation [9, 19], automatic event logging [5], and object of focus identification [8]. Our work contrasts these other foci by demonstrating techniques for supporting human based annotation of events that occur in video.

3. INTERVIEWS & COLLABORATION

To gain a deeper understanding of methods, analysis processes, bottlenecks, and types of data needed for effective video annotation software, we maintained an active dialogue with researchers (in Special Education, Speech and Hearing Sciences, and Computer Science) who use video annotation, conducted informal 40 minute interviews with two experienced video coders, and refined functionality through dialog with current users of VCode and VData. Existing tools for video annotation may address a subset of the below described requirements, however, our system more fully satisfies all of them.

R1 *Facilitate Coding Workflow:* The coding workflow consists of; (1) establishing video clips and coding guidelines, (2) intense training of coders and checks for reliability, (3) annotation of videos, (4) weekly reliability checks on annotated videos, (5) repeat 3 and 4 ad infinitum, (6) analyze data in statistical packages. Tools targeting video annotation should attempt to optimize the transition between steps in this workflow.

R2 *Video, Annotations, and Coding Guidelines should be presented in a synchronized manner:* Interviewees described their coding process centering around analog video on a TV-VCR device, annotating in a Microsoft Excel file, and referencing lengthy code guidelines. Due to the visual separation between annotations, source material, and video, coders had great difficulty during reviews.

R3 *Capture Appropriate Data:* Researchers and existing literature indicate that there are different types of data that are collected through the annotation process: counting events/occurences, determining duration of events, assigning levels, values, or ranking to events, performing phonetic transcription, and general commenting [14]. Effective interfaces must provide methods for capturing these conceptually different data types while preserving each of their unique nuances.

R4 *Additional data should be displayed to coders:* Effective annotation tools should allow researchers to provide additional data to coders to aid in their assessment of video; for example, a volume histogram of the current video, sensor/log data collected in tandem to the video capture, or annotations made automatically or from another source. Displaying additional datapoints has shown to increase the accuracy of coded events [6]. Further, annotation software should facilitate the management of multiple video streams to get the most accurate "view" on the session, and thus produce the most accurate data [4].

R5 *Allow multiple forms of playback:* Researchers mentioned that continuous playback is not always the preferred method of analyzing a video. Often multiple modes of playback are utilized; continuous or standard playback, continuous interval playback (play for N seconds, then stop), and skip interval playback (jump N seconds, then stop). This allows the video to be divided in to smaller segments for annotation of events that are more difficult to pinpoint (i.e. when a smile starts or ends) [4]. Though conceptually simple, manipulations of video using a standard VCR was described as "annoying" and "a mess" due to hand eye coordination and repeatability issues.

R6 *Agreement calculations should be easy and manipulatable:* Regardless of agreement technique used, researchers expressed a frustration in attempts to calculate interobserver reliability. Specifically, existing solutions were limited to importing data into a statistical software package for calculation or calculating them by hand. Video annotation tools should provide quick & easy reliability calculations for individual variables, as well as overall.

R7 *Provide functionality for visual, graphical and contextual review of annotations:* In interviews, coders lamented the process of ensuring reliability on a weekly basis; as it consisted of searching through printouts of a spreadsheet for discrepancies. Specifically, by lacking context in this spreadsheet coders found it difficult to recognize what a given coding mark referred to due to the lack of synchronization with video. By providing a visual, graphical way to review annotations (in the context of the video) coders would be better able to justify the decisions, determine the correct solution, and save time identifying the errors.

4. VCODE AND VDATA

VCode and VData are a suite of applications which create a set of effective interfaces for the coding workflow following the above design requirements. Our system has three main components: VCode (annotation), VCode Admin Window (configuration) and VData (examination of data, coder agreement and training). The interaction with VCode and VData is demonstrated in Figures 2-4 in which two coders are marking a video of a child in an experiment, and checking the agreement between their annotations. The reader should note our solution is only one possible implementation of the design requirements, and that these requirements could be applied to improving existing video annotation software.

4.1 VCode

The VCode application (Figure 1) is designed to provide researchers with an effective way to obtain reliable data from an observational research video. By allowing researchers to present multiple video streams in addition to other sensor data (e.g. log data, annotations from other software, or signals recorded by a computer/monitoring device) the coder can make the best annotation decision possible.

Video: To facilitate multiple video streams VCode presents one main video at full size, and a dock with other streams playing in real time. When a docked stream is clicked on, it repositions itself into the main video window, while the video which was the previous focus, scales down to the dock, thus equating visual importance with relative size and visual weight.

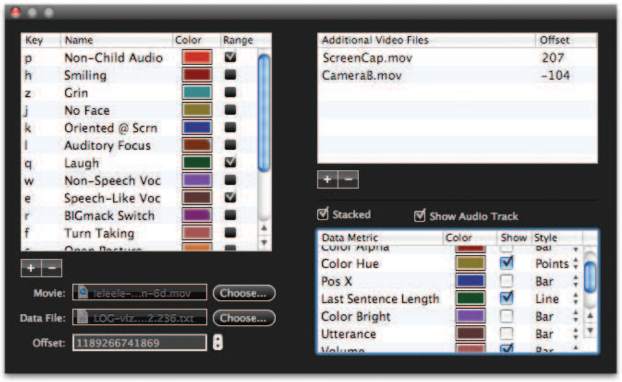
Events: When annotating a video, two different classes of coding events emerge: ranged and momentary. A ranged event is one which extends over a period of time (marking action start and duration). Momentary marks have no duration, and thus represent one specific moment in time. Comments can be attached to any mark, allowing additional observations, levels/ranking, or phonetic transcription (through onscreen phonetic keyboard). Any mark with a comment has a inverted outlines to signify that it has a comment attached. Figure 1 shows a ranged event representing the length of time which a child is making a sound, with additional momentary marks at the start noting other features of the child's state of being).

Timeline: The timeline is the heart of VCode. It is modeled after the moving timeline one might find in a video editing application (e.g. iMovie, Final Cut Pro, etc.). Events, graphically represented by diamonds, appear in a spatial linear fashion to sync with the video. Once an event has been placed on the timeline, it can be graphically manipulated by dragging, clicking, and double-clicking. The standard solution for dealing with large numbers of tracks or variables is to provide a vertical scroll bar or overlay tracks. Rather than limiting the amount of information on screen by scrolling, tracks representing momentary events are "stacked," such that they vertically overlap. This optimizes usage of the screen while still providing enough area for track isolation and selection, even under dense data conditions. Ranged event tracks are unable to benefit from this stacking optimization because of the more complicated interaction for manipulation and thus are vertically tiled. Researchers can present video volume, sensor data, software log data (from Eclipse or Photoshop for example), and even other annotations to the coders. This additional information is presented graphically to the users by bar, line, or scatter plot. This secondary data can allow coders to annotate data captured by other sources than the video streams, as well as provide additional context to their code. For example, should a coder be instructed to mark when a certain noise occurs, he can line the mark up with an audio peek, rather than estimate it and be concerned with reaction time.

Interaction: Annotations can be inserted into the timeline via UI buttons or keyboard hot keys. To optimize the typically complex transport controls we isolated the key activities that coders need execute and provided controls limited to play/pause buttons, coarse and fine grained playhead positioning, and step controls. The three modes of playback outlined in R5 are available.

4.2 VCode Administration Window

To ensure consistent configuration between coders and sessions, all administrative features are consolidated in a single window. The expected workflow is such that a researcher would setup a single coding document with all the variables to be used on all the videos. This template would then be



Key	Name	Color	Range
p	Non-Child Audio	■	☑
h	Smiling	■	□
z	Grin	■	□
j	No Face	■	□
k	Oriented @ Scrn	■	□
l	Auditory Focus	■	□
q	Laugh	■	☑
w	Non-Speech Voc	■	□
e	Speech-Like Voc	■	☑
r	BIGmack Switch	■	□
f	Turn Taking	■	□
	Open Posture	■	□

Movie: leleele-...n-6d.mov
Data File: LOG-viz....2.236.txt
Offset: 1189266741869

Additional Video Files	Offset
ScreenCap.mov	207
CameraB.mov	-104

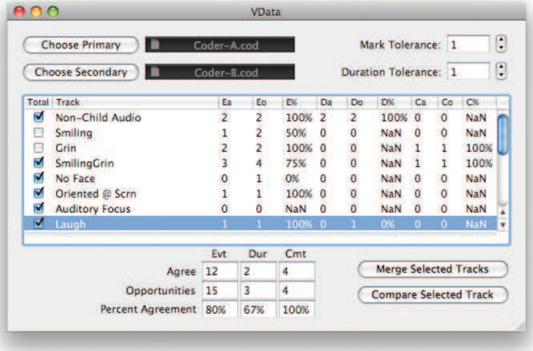
☑ Stacked ☑ Show Audio Track

Data Metric	Color	Show	Style
Color Alpha	■	□	Bar
Color Hue	■	☑	Points
Pos X	■	□	Bar
Last Sentence Length	■	☑	Line
Color Bright	■	□	Bar
Utterance	■	□	Bar
Volume	■	☑	Bar

Figure 2: The code is specified in the Administration Window along with the different video angles, screen capture, and log data.

duplicated (with media and log files inserted for each trial). The main task the Administration Window (Figure 2) is to facilitate is the creation of tracks, used to code data. Researchers can add, remove, and reorder tracks which appear in a list format. The name, color and hot key of each tack can be set through this list presentation. Tracks can be enabled as ranged events through a check box in this interface. The Administration Window is also where a researcher specifies videos and data file to be coded, as well as secondary data for contextual annotation. These elements are specified and synchronized through a drag and drop interface, all of which is hidden from the coder to prevent configuration corruption.

4.3 VData



VData

Choose Primary: Coder-A.cod
Choose Secondary: Coder-B.cod
Mark Tolerance: 1
Duration Tolerance: 1

Total	Track	Ea	Eo	E%	Da	Do	D%	Ca	Co	C%
☑	Non-Child Audio	2	2	100%	2	2	100%	0	0	NaN
□	Smiling	1	2	50%	0	0	NaN	0	0	NaN
□	Grin	2	2	100%	0	0	NaN	1	1	100%
☑	SmilingGrin	3	4	75%	0	0	NaN	1	1	100%
☑	No Face	0	1	0%	0	0	NaN	0	0	NaN
☑	Oriented @ Scrn	1	1	100%	0	0	NaN	0	0	NaN
☑	Auditory Focus	0	0	NaN	0	0	NaN	0	0	NaN
☑	Laugh	1	1	100%	0	1	0%	0	0	NaN

	Evt	Dur	Cmt
Agree	12	2	4
Opportunities	15	3	4
Percent Agreement	80%	67%	100%

Figure 3: Later, analysis is performed on independent codings of the same video. A track with low agreement can be reconciled by viewing the results of two coders side-by-side in VCode, thanks to the capabilities of the VData analysis tool.

Critical aspects of the video coding workflow (training, reliability, and accuracy) revolve around demonstrating agreement between coders. VData (Figure 3) is a separate executable application specifically targeted to aid researchers in training and agreement analysis of coded data produced in VCode.

Multi Coder Analysis: By loading two VCode files into VData, tracks are automatically loaded into the main data table which presents opportunities, agreements, and percentage agreement. For each event (momentary or ranged) an opportunity is said to occur when the primary coder makes a mark. If the secondary coder also makes a mark within a specified short interval, the marks are said to agree. A percentage is calculated from $\frac{agreements}{opportunities}$ for easy interpretation. A tolerance variable is also present to (1) accommodate for variability in the mark placement by the coders, and (2) recognition that there is no quantization of marks beyond the granularity of the millisecond timescale, a property of the system. VData also provides agreement for ranged events and annotations in a similar fashion. It is not uncommon for multiple tracks or variables to be measuring slight variations on a theme (e.g. smiling vs. large smile vs. grin), thus VData implements a track-merging feature which allows opportunities on two distinct tracks to be treated indistinguishably. For a holistic view, researchers can select tracks to be added into a total agreement calculation. In other words, if analysis determines that a single track is not reliable or it is determined that a given track will not be used in the future, it can be easily excluded from the total agreement calcuation.

Conflict Resolution & Exporting: We have optimized coder training and reliability analysis by providing a graphical mechanism to directly compare annotations of two coders. VData can create a VCode session containing specific tracks of two individual coders for side-by-side comparison. The visual, side by side, representation of the data makes it easy to recognize systematic errors in context and detect differences between two coders markings. This reduces the time necessary to locate discrepancies and discuss the reasons why they might have occurred. It is necessary to keep records of these agreement analyses performed with VData by text export. Maintaining export at each stage of the process provides additional transparency and maintains traceability of results that come out of the system.

4.4 Implementation

Our system was implemented in Objective-C using the Cocoa Framework for Mac OS X 10.5. VCode supports all video formats and codecs supported by QuickTime to enable wide compatibility with available video files.

5. MEETING REQUIREMENTS

To ensure Video, Annotations, and Coding Guidelines are presented in a synchronized manner, VCode provides a unified interface containing the target video, a timeline with graphically represented annotations (ranged event, momentary event, or comment depending on data metaphor), additional tracks of signal data (to increase accuracy), and a list of coding guidelines which place marks and stand as a visual reminder. Three forms of video playback (continous/standard, interval playback, skip interval playback) are available via check boxes on the main VCode window to allow easily switching between modes of playback.

VData provides a dynamic interface for real time calculation of multiple agreement values to facilitate easy and dynamic agreement calculations. Through the transparent calculation process, researchers can see both the raw data, and the percentages side by side for easy judgements about the reliability of data collected. Upon request a visual, graphical and contextual review of annotations for both agreement review and training is supported.

Finally, the Coding Workflow is encouraged through VCode's template model in conjunction with the separate VCode Adminstration Window for easy set up and configuration. Training, data collection, and inter-coder agreement are enabled through a tight collaboration between annotation environment and agreement analysis. By consistently providing data export, researchers can be assured that any information annotated by coders can be easily extracted and exported into the statistical analysis tool of their choice.

6. INITIAL REACTION

To evaluate our system in a cursory fashion, we conducted an informal series of interviews with several coders that used our system during the course of an independent study. Analysis using VData showed inter-observer agreement was good and provided valuable coded data for the study. In general, comments from the coders were positive, especially when comparing the VCode system to non-computerized methods. One coder wrote: "The software was easy to use in general, and cut down on coding time." Several features of VCode stood out in their comments; color coding of tracks provided direct linkage between events on the timeline and the description panel, the correlation between files was clear to see during review, sensor data helped anticipate events and accurately code them. It was also noted that the sensor data provided reassurance that what they had noticed in the video was actually correct.

In addition to these positive marks we uncovered several shortcomings of the interface. The seemingly low-resolution bar-graph of volume data left coders unsure where precisely to make their mark. Because the elements of this graph are relatively wide, it appears especially coarse in comparison with the precision with which one may place a mark on the timeline. A spectrogram was suggested as an alternate visualization of the audio data that could help understand sound and video.

Overall, results from these interviews were very encouraging and suggest a more formal study to determine if performance improves in the same way that coders stated that they felt as the tool lowered the amount of time necessary for coding.

7. CONCLUSION AND FUTURE WORK

Video annotation tools can be valuable to researchers by enhancing the annotation process through increased reliability, repeatability, and workflow optimizations. However, many existing solutions do not fully address all the needs of researchers and coders; effective representation of data on a timeline, efficient and robust interaction techniques with video and data, and support for the full video annotation workflow. Our research has provided many contributions in addressing these weak points.

We create a set of design requirements based on existing literature and annotation techniques, interviews with experienced coders, and discussions with researchers in multiple disciplines. Based on these investigations, we implemented a system, VCode and VData, that largely satisfies the requirements we outlined. These systems were then used in ongoing research, and coders were interviewed concurrent with and after using the software, and their reactions were solicited. Our model demonstrates how video annotation software, for many disciplines, can be enhanced to meet the needs of both researchers and coders.

From the reaction of the coders, as well as our own assessment of VCode and VData, we have many directions of possible future work. One avenue is creating a database or networked system in order to facilitate remote access to content, and management of coding objects and assignments for individual coders. It is foreseeable that the system could be extended to a tool to prepare coding files; assist in dividing up raw footage, syncing data to video enmasse, and other automation hooks. This could leverage some of the other existing work in automatic video segmentation. Lastly, we hope to address some of the concerns of our coders, including creating a richer set of data visualizations.

8. ACKNOWLEDGMENTS

We would like to thank NSF for their support of this work (NSF-0643502).

9. REFERENCES

[1] *Annotations as multiple perspectives of video content*, Juan-les-Pins, France, 2002. ACM.

[2] *The Family Video Archive: an annotation and browsing environment for home movies*, Berkeley, CA, 2003. ACM.

[3] *Fluid interaction techniques for the control and annotation of digital video*, Vancouver, Canada, 2003. ACM.

[4] *A Coding Tool for Multimodal Analysis of Meeting Video*, Montreal, Quebec, Canada, 2004. IEEE.

[5] *Creating Multi-Modal, User-Centric Records of Meetings with the Carnegie Mellon Meeting Recorder Architecture*, Montreal, Quebec, Canada, 2004. IEEE.

[6] *Work-in-progress: VACA: a tool for qualitative video analysis*, Montreal, Quebec, Canada, 2006. ACM.

[7] K. J. Berry and P. W. Mielke. A generalization of cohen's kappa agreement measure to interval measurement and multiple raters, 1988.

[8] M. Bertini, A. Del Bimbo, R. Cucchiara, and A. Prati. Applications ii: Semantic video adaptation based on automatic annotation of sport videos. In *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval MIR*, 2004.

[9] S.-C. Chen, M.-L. Shyu, W. Liao, and C. Zhang. Scene change detection by audio and video clues. www.cs.fiu.edu/~chens/PDF/ICME02_Video.pdf.

[10] J. Clifford and G. Marcus, editors. *Writing Culture*. University of California Press, Berkeley, California, 1986.

[11] A. J. Conger. Kappa reliabilities for continuous behaviors and events, 1985.

[12] S. B. Group. Studiocode business group - supplier of studiocode and stream video analysis and distribution software. http://www.studiocodegroup.com, 2007.

[13] A. Johnson. About vprism video data analysis software. http://www.camse.org/andy/VP/vprism.htm, 2007.

[14] M. Kipp. Anvil - the video annotation research tool. http://www.anvil-software.de, 2007.

[15] B. J. Leadholm and J. F. Miller. *Language Sample Analysis: The Wisconsin Guide*. Wisconsin Department of Public Instruction, Madison, Wisconsin, 1995.

[16] L. Lee. *Developmental Sentence Analysis*. Northwestern University Press, Evanston, IL, 1974.

[17] S. S. LLC. Salt software. http://www.saltsoftware.com, 2007.

[18] Noldus. The observer. http://www.noldus.com/site/doc200401012, 2007.

[19] D. Ponceleon and S. Srinivasan. Automatic discovery of salient segments in imperfect speech transcripts. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 490–497, New York, NY, USA, 2001. ACM.

[20] K. S. Retherford. *Guide to Analysis of Language Transcripts*. Thinking Publications, Eau Claire, Wisconsin, 1993.

[21] K. L. Rosenblum, C. Zeanah, S. McDonough, and M. Muzik. Video-taped coding of working model of the child interviews: a viable and useful alternative to verbatim transcripts?, 2004.

[22] S. Soft. Annotation. http://www.saysosoft.com/, 2006.