VIDEO DATA EXPLORATION FOR FILM PRODUCTION AND USER STUDY ANALYSIS

by

Jake Seigel

Submitted in partial fulfilment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
March 2012

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled “VIDEO DATA EXPLORATION FOR FILM PRODUCTION AND USER STUDY ANALYSIS” by Jake Seigel in partial fulfilment of the requirements for the degree of Master of Computer Science.

Dated: March 8, 2012

Supervisor: ___________________________________

Readers: ___________________________________

____

DALHOUSIE UNIVERSITY

DATE: March 8, 2012

AUTHOR: Jake Seigel

TITLE: VIDEO DATA EXPLORATION FOR FILM PRODUCTION AND USER STUDY ANALYSIS

DEPARTMENT OR SCHOOL: Faculty of Computer Science

DEGREE: MSc CONVOCATION: October YEAR: 2012

Permission is herewith granted to Dalhousie University to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions. I understand that my thesis will be electronically available to the public.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

The author attests that permission has been obtained for the use of any copyrighted material appearing in the thesis (other than the brief excerpts requiring only proper acknowledgement in scholarly writing), and that all such use is clearly acknowledged.

____
Signature of Author

Table of Contents

List of Figures

Abstract

This work provides visualizations for digital video management in two different problem domains that have significant commonalities: digital film production (“Bin Explorer”) and user study interview analysis (“Interview Explorer”). The system uses speech recognition software to align a written script to a collection of video footage. For the digital film domain, this means that film editors can efficiently scan through large video “bins” in a meaningful way. Alternatively, the modified version of the software, “Interview Explorer”, offers a novel approach to interview analysis using speech recognition. Additionally, a formative evaluation of the software was conducted. Feedback collected from the participants showed that participants thought the tool offered a better alternative to scanning through the raw interview recordings.

Acknowledgements

I would like to thank my family for supporting me. Also, I would like to thank my supervisor, Dr. Stephen Brooks for keeping me on track, and Dr. Kirstie Hawkey for her guidance in studying humans.

Chapter 1

Introduction

Video management can become an issue in digital film post-production as well as user study interview analysis. Organizing collections of video files can become unwieldy when they are fairly large and the video content lacks distinctiveness. Having to scan through each video file while keeping track of what specific content is covered can quickly become taxing.

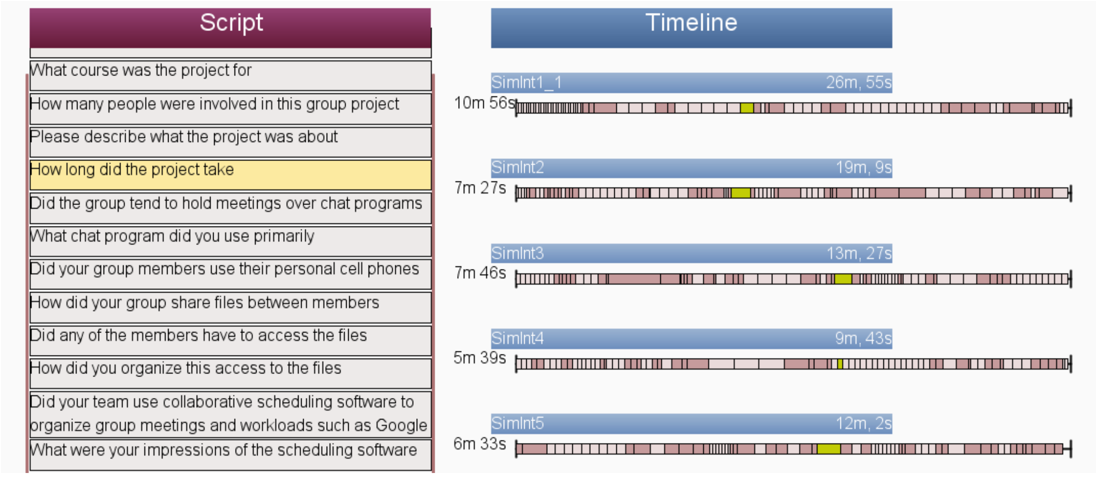
During film post-production, video editing involves sifting through large collections of raw footage stored in folders called "bins". While the footage files in these bins are usually logged, meaning the filenames are tagged with the scene, shot, and take numbers, this provides limited information. The editor must still sort through large volumes of clips while maintaining a list of what parts of the film script each one covers, and ensuring a consistent contrast ratio between cuts (the ratio between the luminance of the brightest colour in an image and the darkest). For that reason, one of the problems this software attempts to address is the lack of organizational assistance during traditional digital film editing.

In this thesis, we present Bin Explorer; a software suite that provides film editors not only with an assortment of media organization tools, but also tools to aid in the clip selection process. The problem of digital media organization is addressed by Bin

Explorer through integrating zoomable interface techniques. In addition to this, the clip selection problem domain is addressed by offering a method to align a film script to the video footage and a visualization that allows exploration. The system is initially reported in [1]. Users are able to import Final Draft® [2] film scripts into the system for the purpose of synchronizing the dialogue text to the lines spoken by actors in the film footage covering each scene. The software uses speech recognition to detect the actors' voices and log the times when they say their lines. To make up for any shortcomings of the first pass of speech detection, a series of error removal and timeframe estimation algorithms are implemented which greatly improve the results.

The results of this script alignment to the footage bin are then displayed in an interactive timeline visualization. This visualization allows the entire bin to be examined, providing the opportunity to efficiently compare the all of the footage where specific lines are spoken. This data can be explored at will and the footage can be scanned through in a manner more meaningful than traditional approaches allow.

The timeline visualization represents each video file with a horizontal array of colour-coded boxes. Each box in the timeline symbolizes a line of spoken dialogue from the footage. The boxes in these timelines are interactive and they connect to the script channel in the workspace. Whenever a section of the timeline is selected, the corresponding line from the script that it represents is also highlighted. In addition to this, the corresponding dialogue in all other timelines is also highlighted, allowing each video take of that footage to be compared as illustrated in figure 1.1.



What course was the project for

How many people were involved in this group project

Please describe what the project was about

How long did the project take

Did the group tend to hold meetings over chat programs

What chat program did you use primarily

Did your group members use their personal cell phones

How did your group share files between members

Did any of the members have to access the files

How did you organize this access to the files

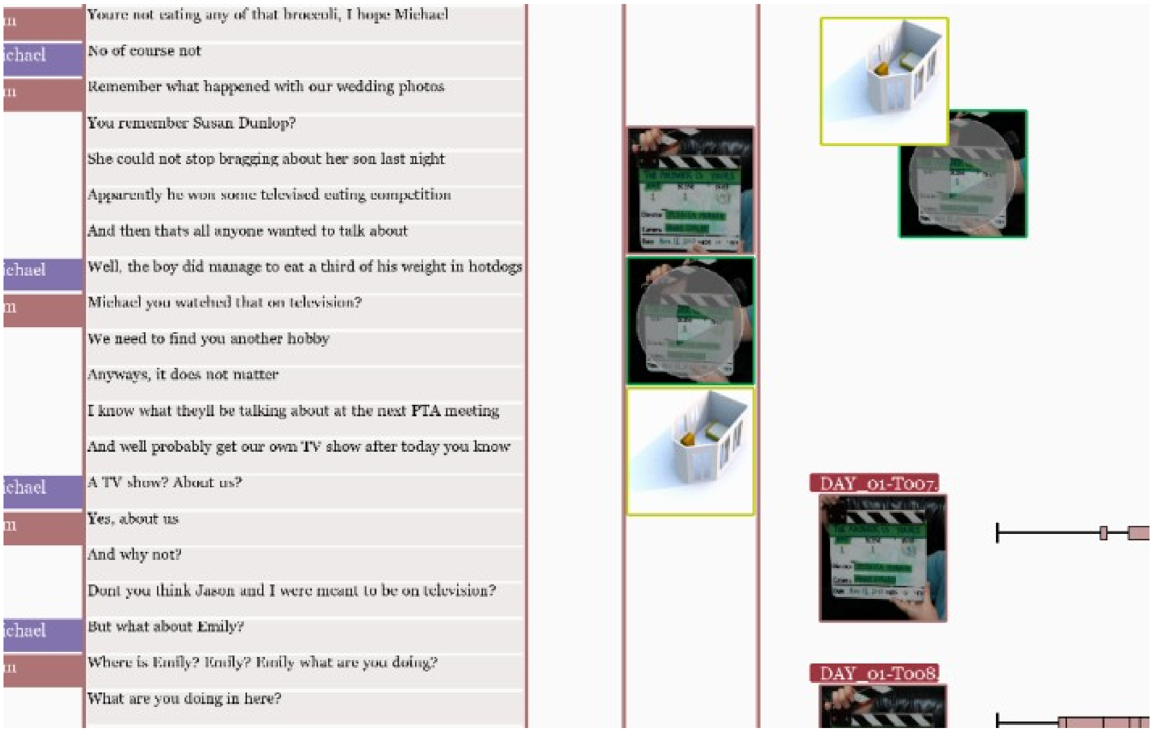
Did your team use collaborative scheduling software to organize group meetings and workloads such as Google

What were your impressions of the scheduling software

Figure 1.1: An illustration of the interactive timeline visualization. The highlighted dialogue from the script channel (*left*) corresponds to boxes on the timelines (*right*).

In addition to the timeline visualization, this software also offers Axis Explorer; a visualization similar to the neighbourhood explorer idea proposed by Spencer [3]. This diagram utilizes a series of axes to represent ordinal data. The data on each axis is sorted according to whatever plug-in is associated with it. While only a limited number of plug-ins are described in this thesis, the extensibility of the architecture allows new plug-ins to easily be created to meet whatever needs the film editors may have.

Furthermore, this system can also be used for the organization of pre-production media. Collections of media such as images, video files, and even floor plans of sets can be imported and organized within a unified zoomable interface. Figure 1.2 shows how the general interface can be used to manage these collections of media panels. This media is represented as a series of panels which can be linked together using collapsible channel widgets. Channels can be scrolled through and manually organized to allow a large amount of media to be managed.



m	Youre not eating any of that broccoli, I hope Michael
ichael	No of course not
m	Remember what happened with our wedding photos
	You remember Susan Dunlop?
	She could not stop bragging about her son last night
	Apparently he won some televised eating competition
	And then thats all anyone wanted to talk about
ichael	Well, the boy did manage to eat a third of his weight in hotdogs
m	Michael you watched that on television?
	We need to find you another hobby
	Anyways, it does not matter
	I know what theyll be talking about at the next PTA meeting
	And well probably get our own TV show after today you know
ichael	A TV show? About us?
m	Yes, about us
	And why not?
	Dont you think Jason and I were meant to be on television?
ichael	But what about Emily?
m	Where is Emily? Emily? Emily what are you doing?
	What are you doing in here?

Figure 1.2: A general view of our system. An imported Final Draft® script in a channel (*left*). A media channel (*center*). Several loose media panels (*top right*). The timeline visualization (*bottom right*).

The interface to our media management system also provides users with two non-traditional zooming controls to assist in both navigation and object zooming as some projects can involve very large workspaces. The first of these techniques is the speed-dependent zooming technique for aiding workspace navigation [4] [5]. The level of zoom within the workspace is dependent on how fast the user is panning through. For example, if the user rapidly scrolls through the workspace then the camera zooms outwards to maintain perceptual consistency. This technique is suitable for almost any situation where large documents must be navigated while maintaining perceptual consistency. The second of these zooming techniques is the CycloStar technique for facilitating object zooming [6]. Drawing a circle with the cursor around a specific position in the workspace will cause the viewport to zoom into that position. This technique is useful for zooming into particular areas of the workspace while not losing track of them during the process.

In addition to dealing with problems in the digital film production domain, a second domain is addressed with a repurposed version of the software; user study interview analysis assistance. This variant of the software, renamed “Interview Explorer”, intends to aid research teams as they perform the tedious and time-consuming process of user study interview analysis. Typically, if researchers need to refer to a previously recorded user study interview recording, they would have to scan through the entire file sequentially. This task can become difficult in situations where clarification is required or when other members of the team need to be brought up to speed on the content.

Interview Explorer aims to streamline this process and allow for better methods of data exploration. The speech recognition tools can be used to segment interview recording files and the timeline visualization provides the means for exploration. While this process is very similar to those used for the digital film production domain, several modifications have been made to the software to greatly improve the results of the analysis.

Lastly, a formative evaluation of Interview Explorer’s feature set has been conducted. The usability of the interface as well as the effectiveness of the interview analysis tools was the focus of this evaluation. Participants were able to successfully complete a list of tasks ranging from locating areas of interest within a corpus of simulated interviews to finding trends between interviews. The feedback and observations suggested the need for several changes to the general interface. However, the participants collectively agreed that the system provided them with an important tool for analyzing interview recordings. Further details of this formative evaluation are discussed in section 7.

The remaining content of this thesis is presented as follows. Chapter 2 provides background of the work related to these problem domains. Chapter 3 describes the general software interface including the zoomable controls, in order to facilitate comprehension of the rest of the work. This also includes the collection of widgets that provide the media organization features of Bin Explorer. Chapter 4 introduces and describes the speech recognition tools, the error removal algorithms, and the prediction techniques that form the core of the system. Chapter 5 illustrates how the visualizations developed in this work are used as solutions for data exploration.

Chapter 6 and 7 discuss the two distinct problem domains: digital film production assistance and user study interview analysis respectively. These chapters examine the differences and similarities of the domains and the modifications to the system for each. Use case scenarios for both applications are in their respective chapters and chapter 7 presents a formative evaluation of the software. Furthermore, each of these chapters discusses future additions to the system for each problem field. Lastly, chapter 8 gives a summary and last discussion about the system as a whole.

Chapter 2

Related Work

Our system is related to several distinct areas of research. Navigation and zooming of large workspaces is a goal of zoomable interface research. Also, several script authoring suites are similar to Bin Explorer as they address problems related to digital film production media management. The Axis Explorer visualization is related to a collection of multivariate data visualizations as well. While a small number of systems exist for the field of qualitative interview analysis, they are only marginally related to Interview Explorer. Lastly, the segmentation features of our system are similar in some respects to a number of content analysis and video summarization systems.

2.1 Zoomable Interfaces

Bin Explorer's user interface features could be compared to several existing zoomable user interface graphics toolkits. But, as there have been a large variety of these zoomable interfaces, only the most pertinent systems will be discussed here. These systems are either pioneering examples of this technology or are directly related in some way.

Pad and Pad++ are examples of early prototypes for demonstrating the feasibility of zoomable interfaces [7] [8]. In addition to this, there are two other zoomable graphics toolkits of note which have succeeded Pad++; Jazz and Piccolo, both supporting more modern languages. The Jazz toolkit offers zoomable features for development in Java [9], while Piccolo provides functionality features for Java as well as the .NET platform [10].

IMapping is an example of an application of the zoomable toolkits for general information structuring [11]. Haller and Abecker compare the “iMaps” created using the system to a whiteboard workspace, where post-it notes can be added. These information pieces can be further nested within each other using the zooming features of the system and sufficient navigation tools are provided for viewing.

Beyond these early systems, additional zooming techniques have been developed in recent years which have been incorporated in this work. The first of these is CycloStar [6], which allows users to draw clockwise circles around a target area to zoom in to. The speed of this zooming action is dependent on the radius of the circle being drawn; larger circles will zoom in slower, and smaller circles are quicker. This zoomable technique allows users to zoom into a specific area of a workspace while maintaining their spatial orientation.

Another zooming technique implemented in this work is Igarashi and Hinckley’s speed-dependent automatic zooming technique for large documents [9][2]. This method will cause the document view to smoothly zoom out when the user is rapidly scrolling through a document. The goal of speed-dependent automatic zooming is to allow a user to navigate a workspace quickly without becoming disoriented. Since the perceptual scrolling speed stays constant, navigating large documents or maps is a simpler task. These zooming and navigation techniques have been included as the production media management features of this system can lead to unwieldy workspaces when the project has a large scale.

2.2 Script Authoring Suites

Script authoring suites are software systems which provide word processing tools specific to script creation for television and film media. There are three of these pre-production systems relevant to this work; Celtx (from Greyfirst Corporation) [12], Final Draft® (from Final Draft Inc.) [2], and Story (from Adobe®) [13].

The first of these systems, Celtx, is a free software system offering tools to not only aid in the creation of screenplays and related media, but also in pre-production media management. Media files can be added to the management interface where metadata such as descriptions can be attached to them. Viewing and modification support is not provided by this system however, and the default software on the computer is used for this task. Furthermore, a scheduling system for managing film shoots is also included, which features report generation options for maintaining control over production.

Final Draft® is considered to be the industry standard for screenwriting tools. Authors are provided with a set of tools for formatting standards. However, since Final Draft® focuses entirely on the writing process, no significant production visualization features or information management tools are provided. The file format for Final Draft® was selected as the primary candidate for screenplay import in Bin Explorer due to its common industry acceptance and XML-based formal.

Lastly, Adobe® Story is one program out of a suite of Adobe® products used for film production. Similar to the previously mentioned script authoring systems, it offers industry-standard formatting options. The strength of Story however, is that it connects to other Adobe® products to add additional features. For example, after using Story® for the creation of a screenplay, the script can be connected to Adobe® OnLocation (a direct-to-disk recording system) to allow the written content to be used as metadata in the video files themselves. This combination of software can also be used to create schedules for filming organized by characters or scene locations. Furthermore, importing a script written in Story® into Adobe® Premiere allows that script to be synchronized to the video file "bin" using speech recognition. The speech synchronization feature allows film editors to create rough cuts of the film.

One of the larger limitations of using the Adobe® Story software suite is that production crews must utilize only Adobe® products to make use of all of the offered features. Unfortunately, it may not be desired or even feasible to do this. Many production studios have their own preferred production tools and may not want to use Adobe® OnLocation during filming or Adobe® Premiere for editing. In addition to this, Adobe® products do not offer any options for comparative analysis of production clips or the management of non-video film production media such as images or floor plans such as those provided by Bin Explorer. And, importantly, their system is not designed for interview analysis and has not been evaluated for this domain.

2.3 Displaying Data Along Multiple Axes

Neighborhood Explorer is a visualization for comparing sets of data items using more than one aspect of comparison [3]. The original system description used a real estate example where a set of houses were compared. The axes in the visualization represent each aspect of comparison between the items; price, area, number of rooms, and distance from vineyard. The diagram limits the number of images represented on each axis by displaying the excess as simple dots.

In addition to this, there exists a number of other ways to present multivariate data in a simple format. Three such examples of these visualizations are parallel coordinate plots, star plots, and a unique system called Dust and Magnets [14]. Parallel coordinate plots use a series of vertical lines to represent variables [15]. Data records are represented by another set of lines which pass through the vertical set. The points of intersection illustrate the values of each variable for every data record. Similar to this, star plots are used for comparing relative values for individual data records (one plot for each record) [16]. Each radii on the plot shows the magnitude of a variable for that record, allowing a simultaneous comparative analysis of data items. Dust and Magnet uses a magnet metaphor to allow users to interactively explore sets of multivariate data [14]. Variables are represented by magnets, which attract or repel points of data with high and low values respectively. This is a compelling metaphor and in theory Dust and Magnets can incorporate an arbitrary number of dimensions. In practice more than three dimensions can lead to an ambiguity of interpretation. Dust and Magnet is loosely related to this work as it resides in the same family of multivariate visualizations as the Axis Explorer diagram. However, this is the only relation that Dust and Magnet has with Bin Explorer.

2.4 Qualitative Interview Analysis

Qualitative interviews are a common tool utilized in HCI research. These can take one of three general forms: structured, semi-structured, or unstructured [17]. Structured interviews provide an interviewer with a strict list of questions, similar to a questionnaire; the interviewer does not stray from this list. On the opposite end of this spectrum, unstructured interviews sometimes take the form of an informal conversation without a script. Semi-structured interviews offer a medium between these two options. While they use a question script, they leave room for additional follow-up questions to probe for more information.

A formal analysis of qualitative interview data is sometimes used to synthesize results from a user study. Some of the more widespread tools for this are Qualrus [18] and Atlas.ti [19]. These tools allow researchers to code textual data from interviews. This means that the collected data can be categorized for analysis purposes. Additionally, transcribed documents from the interviews can be linked to audio and video files of the raw recordings to allow quick referencing.

In addition to these analysis tools, Tagpad is an iPad application for assisting with interviews [20]. It allows the recording of interviews using the device's built-in microphone and provides an interface for manually logging the timing of the questions being asked. When the interviewer asks a new question, they tap the screen to progress the interview. The audio recordings of the interviews can then be uploaded to a computer via Dropbox® [21], where they are segmented into separate files based on the timestamps. Furthermore, the software allows researchers to tag the interviews into categories as they are recorded. While these features are helpful, they are still only marginally better than using a plain notebook and a digital recorder for note taking since no automated tools are provided. Furthermore, it may not be feasible or even desired for the research team to rely on an Apple device for the interview processing as this is the only platform supported.

It is important to note here that at the time of this writing, Tagpad is the only application whose functionality resembles Interview Explorer. One of the differences between it and Interview Explorer is that Interview Explorer can operate on any pre-existing audio recording, where-as Tagpad requires that it be used for the actual recording. Also, Interview Explorer provides automatic methods for video organization and querying.

2.5 Content Analysis and Video Summarization

The work presented in this thesis is also related to video analysis in some respects. The research field of video abstraction has a similar goal of aiming to condense video footage down to important key frames and clips [22]. These systems will only be briefly mentioned however as their goal is only loosely related to this thesis.

A variety of techniques have been researched for this area such as face detection [23], cast listing and video indexing [24]. Related to this, Smith and Kanade [25] and Wolf et al. [26] [26] present systems to detect text in video for browsing and abstraction.

Chapter 3

System Interface Components

The content described in this chapter is composed of several component descriptions. Each of these interface components add functionality to support Bin Explorer's media management goals. Firstly, the Bin Explorer software provides users with a zoomable workspace for managing their projects. Secondly, a variety of media can be stored and organized using a widget called a "media panel". Thirdly, the integration of SweetHome 3D provides Bin Explorer with floor plan functionality. Lastly, the features of the channel interface widget are described.

3.1 Zoomable interface features

A zoomable and scrollable interface provides a home to the various tools within Bin Explorer. Users are free to zoom or scroll through the workspace as they see fit using the typical mouse interaction methods commonly found in computer aided drafting systems [27]. However, as an additional navigational tool in our system, we have also implemented speed-dependent scrolling [4]. This technique will automatically zoom out the view window when the user scrolls rapidly, which helps to ensure that the scrolling speed appears more perceptually consistent to the viewer. This technique was integrated to help aid in workspace navigation of Bin Explorer's media management features.

In addition to this, the CycloStar zooming technique has also been implemented to improve zooming to specific areas of interest within a workspace [6]. Users make use of this feature by drawing circles with the pointer around a focal point of the workspace. The window will proceed to zoom into that section at a rate proportional to the angular velocity of the pointer. This secondary alternative scrolling technique is included in the system as it aids in zooming into specific areas of the workspace, where the speed-dependent scrolling technique only helps in workspace navigation.

3.2 Media Panels

Objects in the software are represented as two-dimensional media panels which can be dragged around the workspace, similar to the three-dimensional, physically-based panels of BumpTop [28]. The media types supported here include images, video files, text files, and 3D floor plans. These panels can also be attached to media channels for the purpose of organization. This is described further in section 3.4.

Figure 3.1 illustrates examples of several media widgets in our system. Image and video panels are represented in our system simply by static thumbnail images (*left* and *center* respectively). Video panels can be invoked, which will play the video file using an embedded version of the VLC media player [29]. Floor plans are represented by a default panel icon (*right*), but when invoked will open a sub-window displaying the 3D file. The file can then be modified or viewed at will using the SweetHome3D software [30], which is described next.

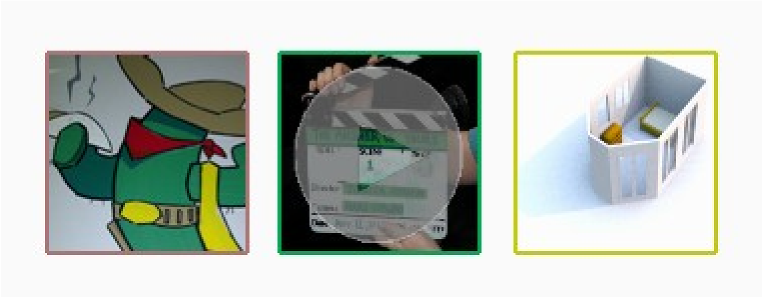
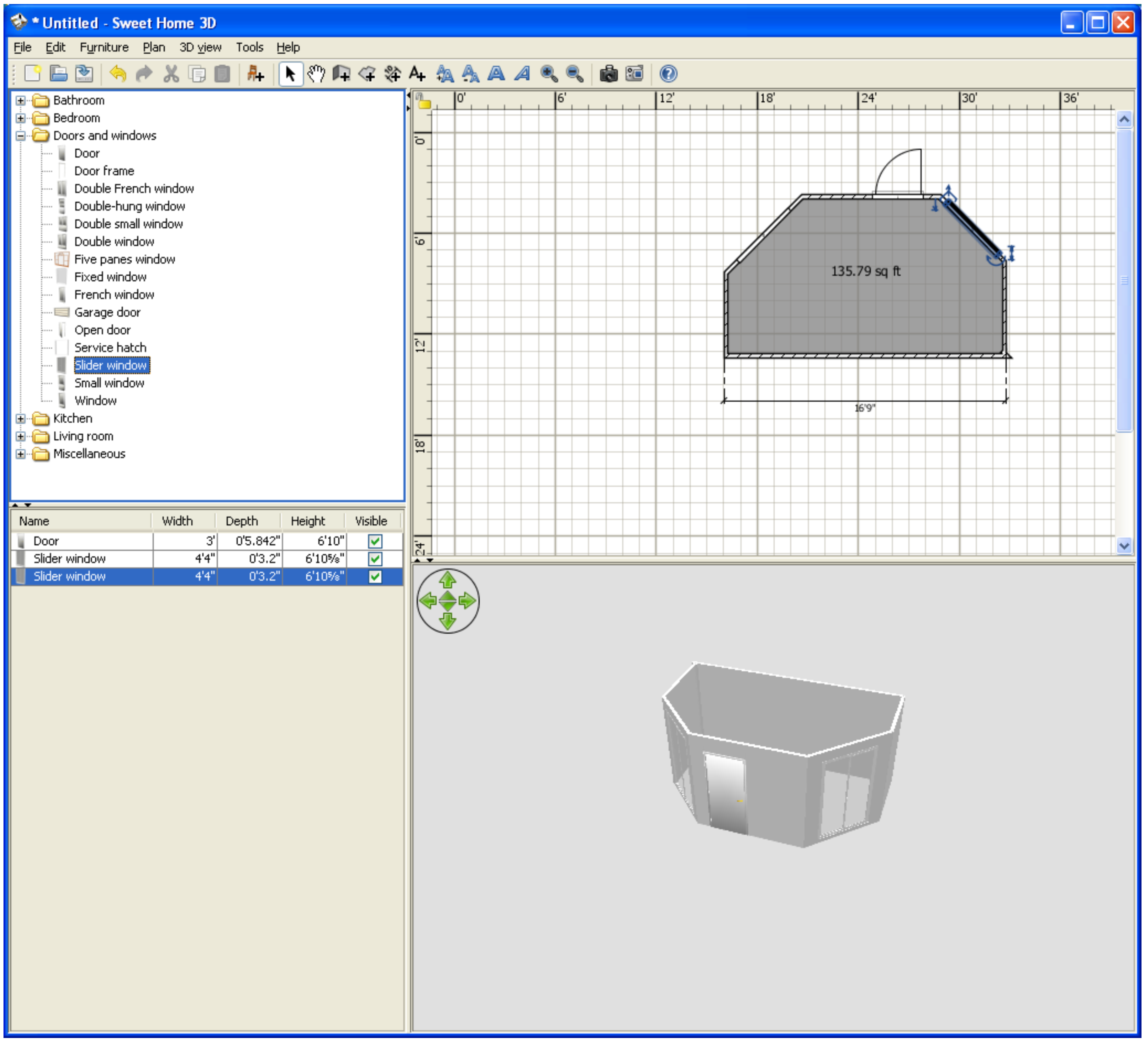


Figure 3.1: Three media panels. An image panel (*left*), a video panel (*center*), and a floor plan panel (*right*).

3.3 Integrating SweetHome 3D

Bin Explorer's floor plan media support is provided through an open-source system called SweetHome3D [30]. This software is included in the Bin Explorer system as it allows pre-production crews to design set layouts and actor blocking. This software provides editors with a series of drag and drop tools for creating floor plans for set design and directorial blocking. Editors are able to draw floor areas and walls directly onto the floor plan workspace. In addition to this, the rooms can be decorated with a collection of prefabricated furniture. The furniture objects can be added to scenes by a simple drag and drop action from the sidebar. Figure 3.2 gives an example of the interface of this software system. The upper-left panel in the figure shows the lists of prefabricated furniture offered by default, while the bottom-left shows the list of furniture imported into the current floorplan. The top-right panel shows the two-dimensional floorplan, and the bottom-right shows a real-time three-dimensional rendering of the current scene.

In addition to the floor plan creation tools, SweetHome3D also provides a high quality, non-photorealistic raytracer. This tool creates stylized renderings of the home defined by the floor plan in any desired viewpoint. Figure 3.3 gives an example of the style of the renderings generated using a small scale floor plan.



Name	Width	Depth	Height	Visible
Door	3'	0'5.842"	6'10"	☑
Slider window	4'4"	0'3.2"	6'10⅝"	☑
Slider window	4'4"	0'3.2"	6'10⅝"	☑

Figure 3.2: The SweetHome3D workspace. A listing of prefabricated, draggable plan components (*top left*). Floor plan workspace (*top right*). The elements within the current floor plan (*bottom left*). A real-time 3D rendering of the current workspace (*bottom right*).

New furniture models can be imported as well to customize floor plan files for their desired purposes. The features in this tool allow it to be used for a variety of purposes ranging from the creation of pre-visualization scenes of film sets, to specifying the layout of a user study locale. In addition to this, a plug-in system allows users to customize their toolset for their own purposes. As an example, this thesis also includes one such custom plug-in. This add-on allows nodes and paths to be specified on a floor plan. This tool could be used on production sets to allow directors to better define actor paths in a scene as well as their timing.

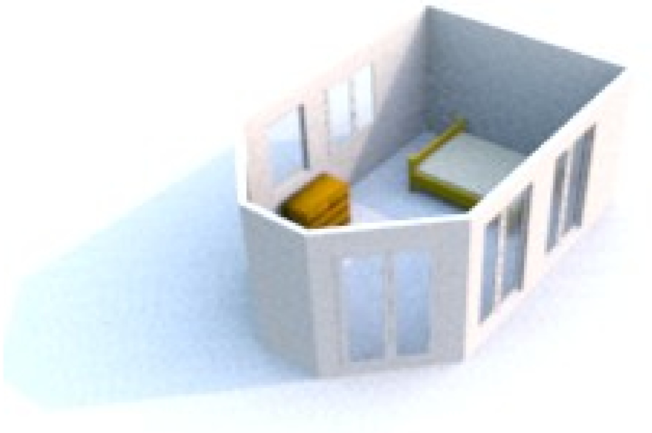
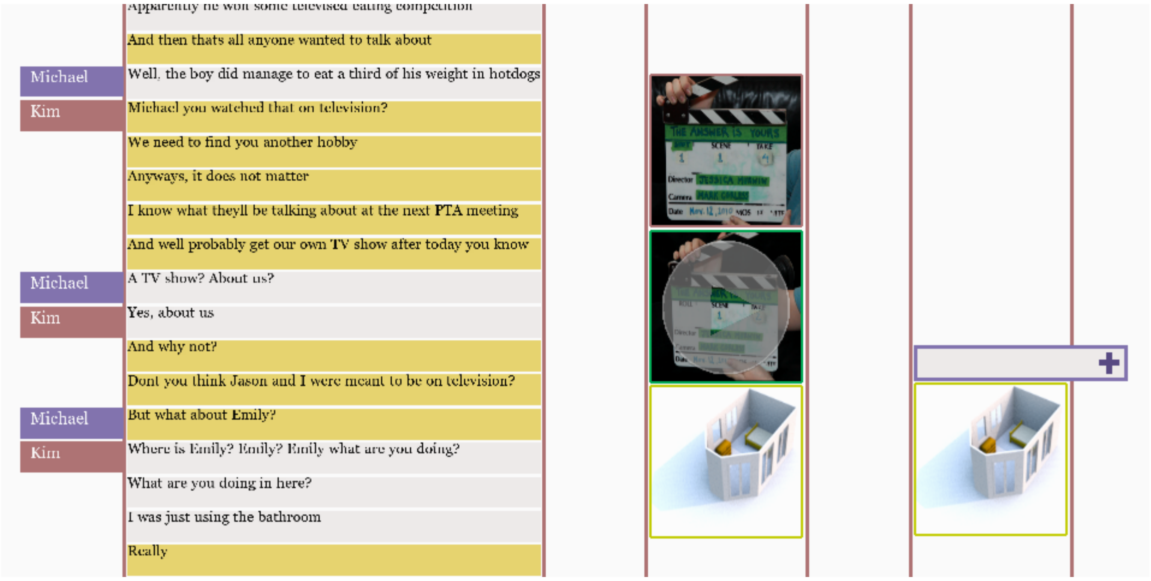


Figure 3.3: An example rendering using the raytracer built into SweetHome3D.

3.4 Channels

The final interface component of this work is the channel widget. This control provides a means to stack and collapse collections of media panels and affords vertical or horizontal scrolling. Any media panel in the workspace can be attached to a channel by dragging it onto the desired position in the channel. This allows channels to be used to create pre-production organizational workspaces.

Furthermore, channels can be collapsed to create more workspace room or to simply focus on specific sets of media panels. Figure 3.4 illustrates how the content in channels can be collapsed. This is performed by dragging the panels within a channel and holding the Shift key. A new "collapse" widget replaces the panels in that channel. The media channel can then be restored to its original state by double clicking the "collapse" widget.



Apparently he won some televised eating competition
And then thats all anyone wanted to talk about
Michael Well, the boy did manage to eat a third of his weight in hotdogs
Kim Michael you watched that on television?
We need to find you another hobby
Anyways, it does not matter
I know what theyll be talking about at the next PTA meeting
And well probably get our own TV show after today you know
Michael A TV show? About us?
Kim Yes, about us
And why not?
Dont you think Jason and I were meant to be on television?
Michael But what about Emily?
Kim Where is Emily? Emily? Emily what are you doing?
What are you doing in here?
I was just using the bathroom
Really

Figure 3.4: A pair of media channels (*center and right*) and an imported Final Draft® script channel (*left*). Channel with three panels before being collapsed (*center*). Channel with the top two panels collapsed (*right*).

Lastly, scripts imported from Final Draft® are added to their own customized channel by default when they are dragged onto the interface as shown in figure 3.4 (*left*). Each line of dialogue in the script receives its own text widget, while the characters' names are reflected by coloured tags attached to the left of the channel. Furthermore, these text widgets can be used to interact with the timeline visualization which is outlined in section 5.1.

Chapter 4

Speech Analysis

This chapter describes the speech recognition engine and error removal processes used by Bin Explorer and Interview Explorer. While this chapter is not necessarily considered a major research contribution, it is of key importance to the workings of this system as a whole. Section 4.1 describes the speech recognition engine used by the Explorer software systems for aligning scripts to the video footage. This process provides a timestamp for each line of dialogue that occurs in a video recording. Section 4.2 discusses the techniques used to detect and remove errors in the results from the speech recognition process. In addition to this, a process is introduced for estimating timestamps for dialogue that was not successfully recognized by the speech engine.

4.1 Speech Recognition

A key aspect of this work is a speech recognition method for analyzing spoken dialogue within recordings. This speech tool is used to detect the lines of dialogue that occur in both the script as well as the raw video footage. Finding these lines from the script is the first step towards aligning that script to the footage bin. For user study interview analysis, the system operates on recordings of structured and semi-structured interviews and the question scripts used during those interviews.

After the film editor imports a Final Draft® script into the Bin Explorer workspace along with one or more clip files, the software will perform the speech analysis. Final Draft® has been selected for this work as it is considered to be the industry standard. It should be noted here that our software only operates on audio files so if a video file is provided, it will pre-process the recording by ripping the audio channel from that file automatically.

Typically in digital film production, a separate audio recording device is used during filming to achieve better quality results than the default microphone built into the camera. However, one issue with using this separate audio file is that it is almost never synchronized to the video footage. This means that even if the script channel is aligned to the audio file, it will be offset from the video footage by an amount unique to every take. Fortunately, acceptible results can still be achieved using only the audio channel from the raw video footage.

The speech recognition engine utilized for the actual audio file analysis is Microsoft's Speech API [31] which is sufficient to achieve passable results. This preprocessing step takes roughly ten seconds for 20 minutes of recording on a multi-core 2.4 GHz computer, depending on the density of lines of dialogue.

Once the speech engine has finished processing the audio file, a list of information about the matched results is returned. Along with a text entry of the recognized line, a timestamp of when the line occurred, and a confidence value are returned as well. The timestamp of each line of dialogue refers to the time in the footage that the line was recognized. This means that only the time that the line finishes is provided, not the start time. The processing techniques described in the following section are required to calculate this starting time. The confidence value supplied is a number from zero to one signifying the strength of the match; a higher value meaning the match is more likely to be true.

4.2 Result Error Removal and Timeframe Estimation

Generally even with the list of expected dialogue for a recording, this speech recognition software still yields less than perfect results (approximately 40-50% of the spoken dialogue is detected), which means more processing must be performed to find a proper timestamp for each line of dialogue in each video file. This lack of accuracy can be attributed to a number of possible reasons; poor quality of recording, variances in annunciation by the speakers, and differences between the script and spoken dialogue. While some of these can be avoided through better quality equipment and better attention to detail from the actors, sometimes these factors cannot be helped.

The procedures described in this chapter are able to handle these missing dialogue matches and false positive results to an acceptable degree for this application, even though the underlying speech recognition system produces less than perfect results. The goal of this error removal phase is to narrow down the predicted time window of script dialogue within the video footage so that we know where the dialogue occurs as well as its length in the clip. It is important to note though that this software may never achieve perfect results. The reason for this is that it depends so heavily on the nature of the audio or video files being analyzed. For example, if there is a large amount of background noise then the speech recognition software may not be able to pick out the target dialogue.

The first technique described here is to simply remove match entries that were tagged with low confidence values by the speech detection engine. By removing match entries with a confidence value below 0.5, matches that would otherwise be considered duplicates or accidental matches are removed. This is not always effective though since the confidence rating could depend more on the quality of speaking or character acting instead of whether or not the line was correctly recognized by the engine.

To reiterate, in addition to the confidence value of the match, the use of the speech recognition tool also produces a list of the times when each line of dialogue occurred within the recording. However, the occurrence time returned by the speech engine refers only to the time when the line finished in the recording. Calculating the actual length, and from this the starting time that a line of dialogue occurs in the recording is the first of several issues requiring additional processing in this phase. In addition to this, more processing must be performed at this point to estimate the timestamps for the lines of dialogue occurring in the recordings which were not detected by the speech engine. Furthermore, the speech recognition engine will generally return false positive matches whose timestamps are wrong and need to be corrected. In general, these issues cannot be fixed by only using confidence filtering.

A more effective approach to improving the dialogue match quality is our sequence analysis algorithm. This approach aims to collect groups of detection matches and use those to perform error removal. Instead of relying on the individual confidence values of the speech results, this algorithm aims to locate adjacent speech matches and compare them to individual matches for the purpose of validating them.

The algorithm begins by first finding a seed match between the speech recognition matches and the dialog in the film script. Let S be the ordered set of script sentences and let R be the ordered set of recognized sentences, where R is a subset of S. We define S_m as our seed match such that $S_m = R_i$, where m and i are indices within each respective ordered set. We then define a sequence, C as:

$$\{C \subseteq R \mid S_{m+x} = R_{i+x}\}, \forall x = 1..n, \text{ where } n \text{ is the size of } C \qquad (1)$$

Figure 4.1 illustrates this process of joining together consecutive matches. Once the set of sequenced matches is detected; they can be used to rule out false speech matches. Since a sequence of recognized dialogue is more likely to be correctly identified than non-sequenced matches, we can rely on them for error correction. The time in the clip when the dialogue sentence is hypothesized to have occurred is compared to the occurrence times of each sequence of matches.

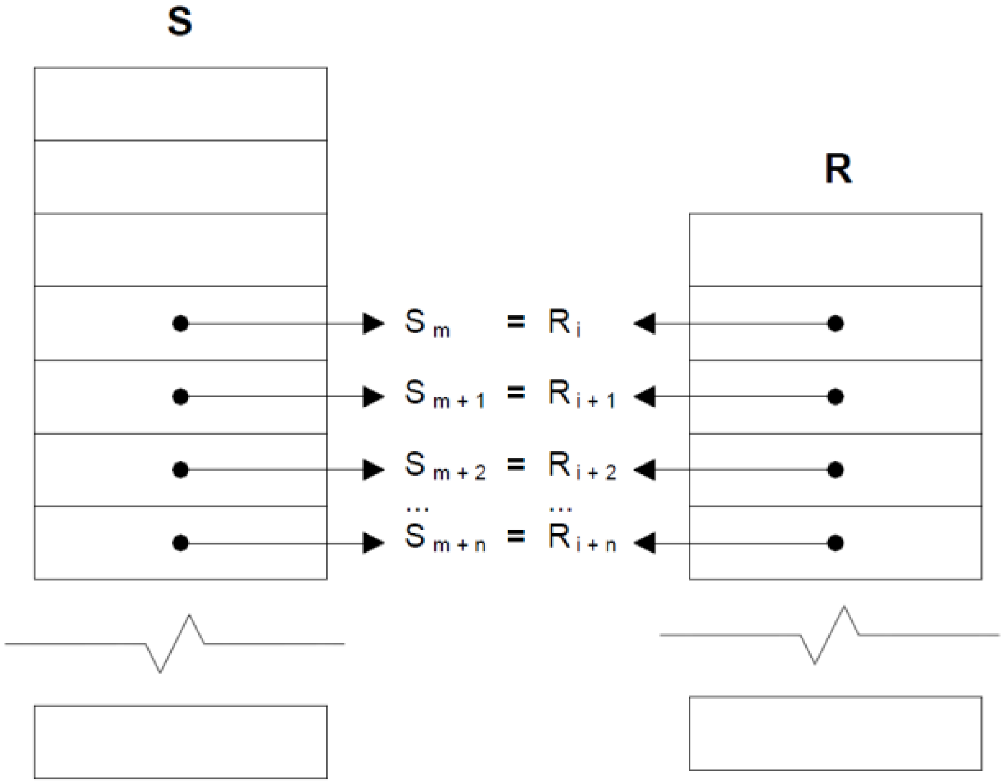


Figure 4.1: Mapping of the set of recognized sentences, *R* to the full set of script sentences, *S*.

We can rule out an entry as false by first checking if the occurrence time for a non-sequenced match, R_i happens after the occurrence time of sequence, *C*. If the sentence in R_i occurs in the script after each sentence in *C*, we can assume that R_i is invalid. Figure 4.2 gives a graphical illustration of this comparison. In one example video shot; out of 28 matched dialogue sentences, twelve were false matches (duplicate and out of order lines). Our system was able to remove every one of these false matches, and so far has had such a consistent level of success with the rest of the clips.

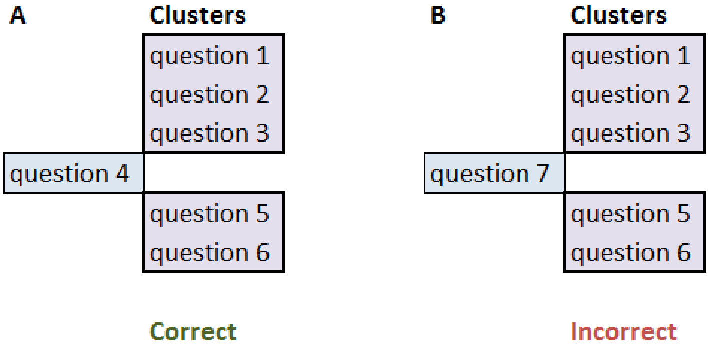


Figure 4.2: An example of comparing individual speech matches with sequences of matches. Example A (*left*) shows question four being compared to the two sequences; it occurs in the script after the top sequence, but before the bottom one. This match is accepted by the system. Example B (*right*) shows question eight in the same comparison; it is rejected as that question was expected after both sequences.

After we verify which lines of dialogue were correctly recognized in the video file, we are able to approximate where the missing lines of dialogue occur within that file. This is accomplished by linearly interpolating the dialogue between the positive matches and sequences. This essentially means that we take a gap of time that we know several lines of dialogue occur in, and we divide up the time equally between each line. Figure 4.3 gives a general illustration of how this process occurs. The dark-coloured sections in the diagram timelines represent the timeframes of lines of dialogue that were successfully detected by the speech recognition engine, and the light-coloured ones are the areas that will be estimated using the linear interpolation. If the light-coloured section in timeline A represents a block of time where four lines of dialogue are expected, then timeline B would be how that time would be divided between the lines. That is to say, the space is evenly divided between the dialogue that is expected to fit inside.

On average this process yields good results, especially for the digital film production application that will be described in chapter 6. When scenes are filmed there is generally a sense of flow for the script, which leads to well spaced out lines of dialogue. This is an ideal condition for the timeframe prediction system as it operates using linear interpolation.

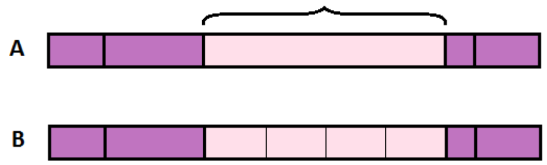


Figure 4.3: A pair of example timelines. Timeline A (*top*) shows the space where four questions are expected to take place. Timeline B (*bottom*) shows how that space is divided between the questions.

The precision of this algorithm can be further improved using the set of false dialogue matches returned by the speech recognition engine. Since we know that the correct matches returned by the speech engine do not overlap with false matches in the clip, we can use the time frames of the false matches to help narrow the window that lines can occur in. This improves the estimation phase of this process as we have a smaller time window for each of the lines we are estimating a timestamp for. At this point, we have a good idea of the start and end times of the spoken lines of dialogue in the scene.

As an example of the general accuracy of the software, the footage for the first scene of a student film was analyzed. The raw footage for the scene spanned close to 20 minutes of footage. The average accuracy of the lines of dialogue predicted by the algorithm was within 1.4 seconds with a standard deviation of 1.02 seconds. Factors that effect this variation are the length of the dialogue and the level of density of character speech in each scene. The larger predicted windows of time were in most instances due to longer periods of time where no dialogue occurred in the video. This would happen while the production crew was preparing for the scene to begin and sometimes in the middle of a scene, between lines of dialogue.

Chapter 5

Visualizations

5.1 Timeline Visualization

After the textual script has been aligned to the recording collection, a visualization is provided to illustrate those aligned matches. Within the digital film domain, the recordings consist of every shot filmed for a single scene. For the interview application, the recording collection is several separate interviews with the same list of questions. While each application of the timeline overlap visualization utilizes a different data set, the underlying interactive tool is essentially the same. However, the specific differences and data sets of the digital film and interview analysis domains will be explained in chapters 6 and 7 respectively.

The timeline tool illustrates the alignment of a predefined script to a collection of recordings. For each recording in the collection, a separate timeline is displayed with interactive sections. Within each timeline, the speech elements from the script are spatially mapped to represent their temporal position within a recording in the form of these interactive sections. The width of each of these sections is determined by the calculated length of the spoken dialogue.

Each section on a timeline is colour coded using a two-tier scheme. Lines of dialogue that were successfully detected by the speech recognition engine are displayed as dark purple sections. Dialogue elements from the script that were only estimated are represented by light pink sections. Figure 5.1 illustrates a timeline where many of the script elements were estimated. The diagram in figure 5.1 also shows one box highlighted in yellow; this means that the corresponding line of dialogue has been selected from the imported script.

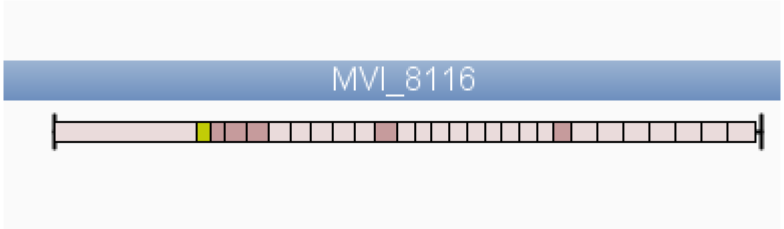


Figure 5.1: A timeline visualization entry. Each section represents a line of dialogue from the script. The gaps in the list of matches are represented by light pink colours and the matches are represented by the dark purple colours. The title of the recording is displayed in the blue text box above the diagram.

Upon clicking on a timeline section, the corresponding lines of expected dialogue from the script channel will be highlighted in yellow. Furthermore, the equivalent dialogue sections from every other timeline in the bin will be highlighted as well.

This aspect of the visualization is meant to aid film editors in finding related clips of video between each video file in the bin. For example, if the user is focusing on footage where a specific line of dialogue is spoken by an actor, then that line of dialogue is simply selected from the script. The section on each timeline representing that dialogue will be highlighted. In addition to this, the timestamps of these highlighted sections are also displayed next to each timeline.

5.2 Axis Explorer Visualization

The next visualization provided by this system is the Axis Explorer diagram; a multivariate analytical tool based on Spence's Neighbourhood Explorer [3]. Axis explorer uses several rotatable axes to represent ordinal data sets for the purpose of interactive exploration and analysis. Each axis in the diagram is used to represent one sorting function, which displays the top (or bottom) matches from the dataset. The data set in this work consists of digital film footage and interview recordings.

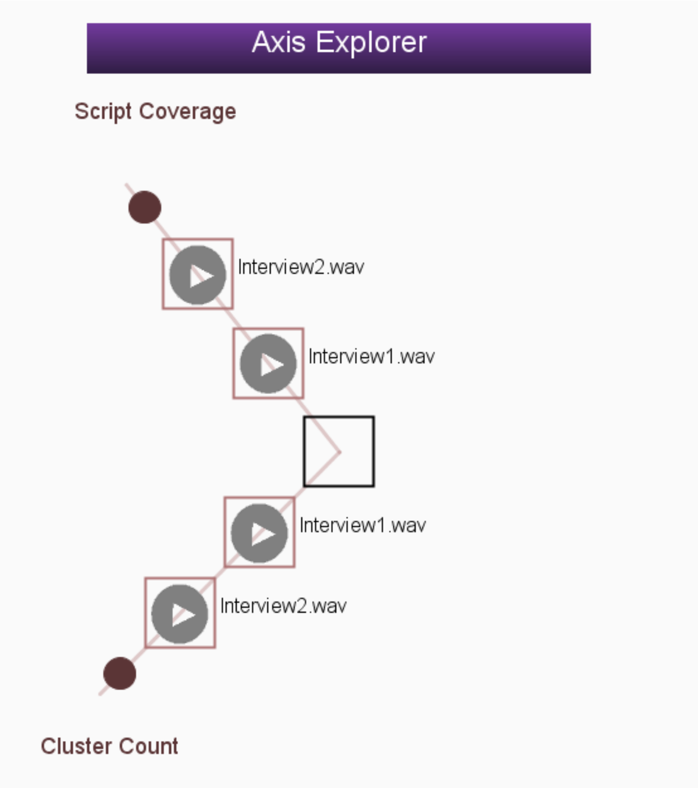


Figure 5.2: A screenshot of the Axis Explorer diagram. The upwards-pointing axis favours videos which feature a larger number of recognized dialogue for the scene. The downwards-pointing axis favours videos which have a larger number of detected dialogue sequences.

An axis in this diagram is a line leading away from the diagram center with media panels attached along it. The angle of the axis has no meaning other than to separate it from other axes. Each of the axes can be rearranged by dragging one of the panels attached to them; this will rotate them around the center of the diagram. The first-ranking items in the dataset are placed closest to the center while the less relevant items are placed further away. Furthermore, to avoid cluttering the diagram, less relevant results are represented by dots on the axis line instead of a panel display. These dots can display more information about the entry if the user focuses on each one. In the future, colour coding these space-saving dots could help to illustrate the distribution of their relevance.

Two basic axes are implemented in our system at the time of this writing as seen in figure 5.2. The first of these is the script coverage axis, which is used to sort the footage bin by the amount of the film script that is covered by each video file. The second of these is the sequence count axis, which is used to sort the bin based on how many sequences were detected by the error removal algorithm described in section 4.2.

A programming interface is provided to users who wish to write plug-ins for their own analysis of the recordings. Each plug-in will add a new self-contained and populated axis to the diagram. The plug-ins are written using the Java programming language and they can make use of any external API required for the analysis they provide.

Chapter 6

Digital Film Production Problem Domain

This section describes the specific details for one of the primary applications of this work: digital film production. The goal for this domain is to aid in the film editing process; specifically, with the selection and analysis of the shots composing a scene. Typically, film editors must manually maintain lists of what sections of the scene each video clip covers. While this is part of the artistic process and will most likely always be a part of film making, the tediousness of these tasks can be somewhat alleviated. By aligning a film script to the raw footage in the bin, scenes can be textually searched to locate areas where specific dialogue occurs. In addition to this, the editor can perform a comparative analysis on the footage using the Axis Explorer diagram.

The first step towards aiding in the shot selection process is to align a film's script to the raw footage. The raw video footage for a scene is generally made up of several takes of the same content and is stored in a folder called a "bin". Each of these takes could cover not only different areas of the script within the scene, but also different angles of the same content. A script alignment refers to the match up of every line of spoken dialogue from the actors in the footage with their lines within the textual script.

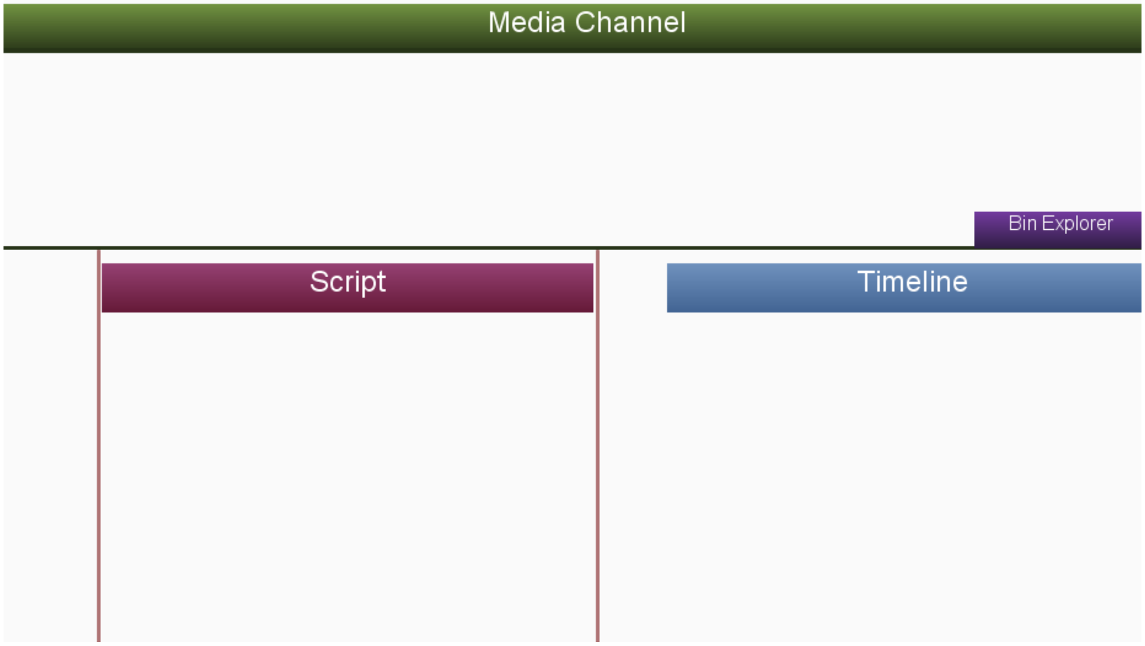
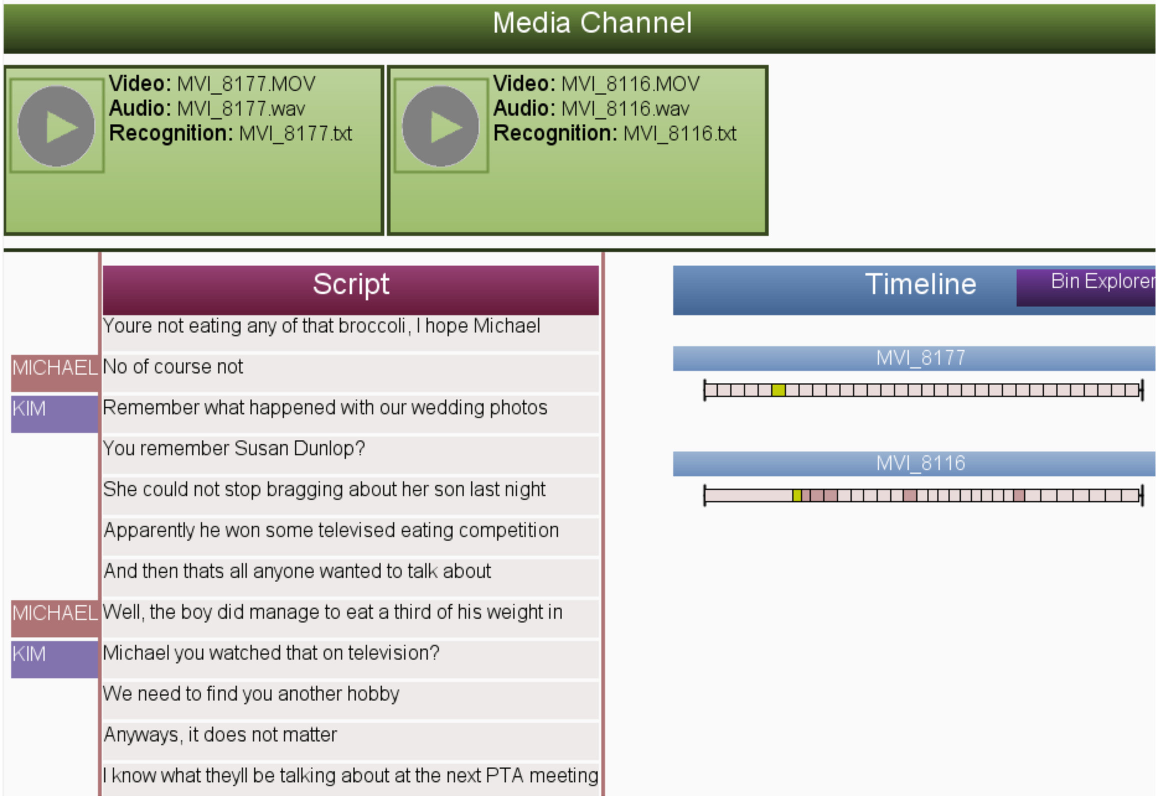


Figure 6.1: An empty Bin Explorer workspace. A horizontal media channel (*top*). A script channel (*bottom left*). A timeline visualization channel (*bottom right*).

For digital film analysis, Bin Explorer offers an explorer version of the software interface to simplify the functionality. Figure 6.1 shows what an empty project this mode in the interface looks like. The green media channel at the top is a horizontal channel which holds the media panels containing the scene footage. The pink script channel holds the script for the scene currently being worked on. The blue timeline channel holds the array of timeline visualizations.



Video: MVI_8177.MOV
Audio: MVI_8177.wav
Recognition: MVI_8177.txt

Video: MVI_8116.MOV
Audio: MVI_8116.wav
Recognition: MVI_8116.txt

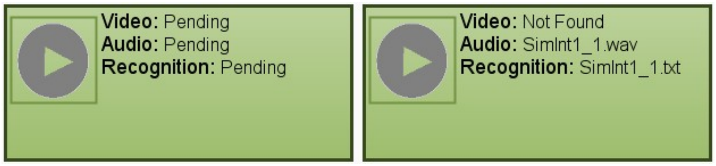
	Youre not eating any of that broccoli, I hope Michael
MICHAEL	No of course not
KIM	Remember what happened with our wedding photos
	You remember Susan Dunlop?
	She could not stop bragging about her son last night
	Apparently he won some televised eating competition
	And then thats all anyone wanted to talk about
MICHAEL	Well, the boy did manage to eat a third of his weight in
KIM	Michael you watched that on television?
	We need to find you another hobby
	Anyways, it does not matter
	I know what theyll be talking about at the next PTA meeting

Figure 6.2: A populated workspace containing two video files.

Figure 6.2 shows a populated workspace. The green media channel is populated with two files of scene footage. Each entry lists the name of the video file, the ripped audio channel file, and the file that holds the results from the speech recognition stage. The script channel is populated with a script from a student film. On the left-hand side of the channel, the characters' names are listed in colour coded boxes.

After adding a script to system, the footage bin can be imported by dragging the folder onto the interface. The footage will be processed automatically by the software; the audio channels will be ripped from the video files and the speech recognition and error removal systems will perform their operations to align the script to the bin. The status of this process is updated in each box in the green media channel, as illustrated in figure 6.3.

Once each phase of the processing is completed, the text entry for each part changes from "Pending" to the name of the newly created file for that step.



Video: Pending
Audio: Pending
Recognition: Pending

Video: Not Found
Audio: SimInt1_1.wav
Recognition: SimInt1_1.txt

Figure 6.3: Examples of the panels attached to the horizontal media channel at the top of the workspace. The status of the panel entries when the recording is being processed (*left*). The status of the panel entries after the recording has been processed (*right*).

Aligning film script to the raw recorded footage requires the speech analysis software to operate on each file in the bin using that script. To reduce the chances of false recognition matches, it is suggested that only the section of the script relevant to the current scene is used. This is an acceptable requirement of editors as a shot will almost never cover more than one scene. Once the speech software is finished processing the film footage, it will return a list of the dialogue lines detected in each clip, which is then processed with the sequence analysis algorithm described in section 4.2.

It should be noted here that in order for the movie script to be aligned properly the writers must properly tag the character names for each line within Final Draft®. This is not an unreasonable request however, since this is a common practice at the professional level. In addition to this, film crews typically include a script manager who will update the script as minor changes are made to it. This is important, as having an up-to-date script during the speech recognition phase will yield better results.

The Axis Explorer visualization provides professional and amateur film editors with a way to compare and sort their footage in a meaningful way. Each axis represents a different method of comparison which can be customized and manipulated by the user. The plug-in system allows editors to tweak the axis comparisons for whatever organization scheme or query they require. This visualization is activated by double clicking the “Explorer” button (seen in figure 6.1), which will replace the timeline visualization with the Axis Explorer diagram. Section 6.2 discusses several plug-ins planned for future development. The most relevant of the plug-ins suggested here is the scene composition tool, which could be used to automatically determine the camera angle of scenes.

6.1 Use Case Scenario: Student film project

Here we present a short use case scenario for our system. The data set is comprised of raw movie footage from a digital film production at a university-film school. To demonstrate our system, we will focus only on clips covering the first scene of the film. Each of these files is over one gigabyte in size and spans two to three minutes on average. The footage bin also provided audio files recorded using a higher quality microphone than the camera’s built-in device. However, the lower quality audio channel from the video footage was still used here as it is properly synchronized. This means that the audio properly lines up with the video, which is generally not the case when a separate audio device is used.

The analysis process begins by dragging the Final Draft® script for the film onto the window of our system along with several of the clips in our scene from the bin. This adds the list of clips to the media channel and the system begins processing the content. The speech recognition system analyzes each file and returns a list of dialogue matches and their occurrence times. Once the analysis is complete, each entry is listed in the timeline visualization. Not all of the video clips from a scene are guaranteed to span the same parts of the script however. For example, some shots could be cut short due to on-set issues with the actors, or may be taken from different angles.

At this point, the film editor can analyze the video files for the purpose of shot selection for the final movie. If the editor wishes to select a subset of the videos containing a specific line of dialogue for part of a scene; it is selected from the script channel. The videos in which the dialogue has been recognized are displayed and the position in time is highlighted in the timelines for each clip. From here, the Axis Explorer visualization is activated by selecting a video clip to be the center of the diagram. We see on the comparison axis which videos have the most amount of matched dialogue overlap. This means that during editing for a specific part of a scene, we can start our search at this location for clips to cut to.

6.2 Future Work for Digital Film Production Assistance

At present, we provide only a limited number of axes of comparison for the Axis Explorer visualization; however, this system is still under development and further semantic axes are planned. Among these future axes, a scene composition analysis tool and a contrast ratio comparison tool are proposed.

The composition tool is used to compare the general positioning of actors in a scene to determine if the shot angle is the same or not. The footage of a scene can be spread over many takes from several different angles but there is no easy way of sorting the footage without viewing it. The tool would work by applying a basic image segmentation operation to key frames within the video footage. These key frames will be selected using the timeframes of dialogue detected by the speech recognition engine. The actors and objects in the scene would be loosely represented by blobs which will then be compared using a simple pixel wise overlap technique. If the overlap exceeds a certain percentage threshold in each key frame, then the two video clips are assumed to be from the same angle. The footage in the bin can then be organized using the Axis Explorer visualization by putting shots from the same angle in the same axis.

The proposed contrast ratio tool would work in a similar fashion; key frames would be selected and subsequently have their contrast ratios analyzed. A contrast ratio is the ratio between the luminance of the brightest colour in an image and the darkest. The contrast ratios of the key frames detected within each video file are then compared to ensure the clips maintain continuity. More key frames would have to be selected than the scene composition tool required in order for the system to achieve a higher level of quality.

One future improvement for the speech analysis system is to use the quiet times within the recordings to improve the quality of the detection results. By logging these times, the software could be able to more accurately predict the occurrences of lines of dialogue. This will be accomplished by using the timeframes from these empty sections along with the sequence analysis algorithm to achieve smaller estimation time windows. The results of this improvement are not guaranteed however; this technique would be susceptible to background noise in the audio channel.

Another feature that might be desired in the future by digital film production crews is a method for automatically aligning the high quality audio channel to the video footage. This lack of alignment makes it generally undesirable to use the higher quality audio for the script alignment as the alignment will not be relevant to the video recording. While this issue could potentially be alleviated by allowing editors to input the offset difference between the audio and video channels, this would have to be done separately for each file.

Lastly, the dialogue estimation method could be modified to provide more exact predictions of timeframes. Currently a linear interpolation scheme is utilized to divide blocks of time evenly between the lines of speech that are said to take place in their boundaries. Instead, this distribution could rely on the length of the dialogue in the script. For example, sentences that are lengthy would be given a larger time window than those that could be comprised of just a few words.

Chapter 7

Interview Analysis Problem Domain

In addition to the digital film production problem domain, our system can be applied to interview analysis as well. Analyzing interviews can be a monotonous and lengthy task. While some researchers rely entirely on handwritten interview notes, it is sometimes preferred to also maintain audio or video recordings of the interviews. These recordings may need to be re-examined at a later point in the analysis for the purpose of response clarification or simply for members of the research team who were not present during the interview recording. Typically, revisiting these recordings could mean scanning through each interview in real time to avoid missing important responses or simply to locate specific areas of interest. This process can take a great deal of time depending on the number of the interview recordings and the length of each one.

Furthermore, in some cases the recordings are sent to external parties for transcription. This can take a significant amount of time in itself, but the quality of the returned transcripts are sometimes found wanting. In some cases this means that members of the research team will have to review the files to confirm responses or even correct flaws. None of these situations are ideal for anyone taking part in these research projects.

Therefore, one of the goals of the Interview Explorer software is to aid in segmenting qualitative interview recordings with the hopes of improving the efficiency of the analysis process. This system offers an alternative to researchers having to scanning through each interview recording while guessing where their specific points of interest occurred. Furthermore, when the researchers are forced to sift through large amounts of recordings, there is a chance they will become fatigued. Results from this could lead to anything from the researchers changing their styles of analysis over time, to missing important events in the interviews. This is another problem that could potentially be addressed by this software system, through improving the speed and efficiency of the review periods and allowing for a more engaging data exploration experience.

The fundamentals of this system are nearly identical for this problem domain as with the digital film production domain. However, the changes made to the software for this domain warrant it to be renamed to "Interview Explorer". The primary difference between this problem domain and digital film production lies within the type of content of the interview recordings. For example, for film footage the takes in each scene in a movie are guaranteed to have the same spoken dialogue from the actors. Actors are constrained to a strict film script and rarely venture from it. But if changes occur, a production crew member will be put in charge of updating the script to reflect the new changes.

With interviews on the other hand, questions are sometimes asked in varying forms or even out of sequence depending on the interview type. This is generally the case with semi-structured interviews where follow-up questions are also asked to probe for more information. This is the primary reason for the modifications to the error removal and analysis engines of the original Bin Explorer codebase. The error removal and timeframe estimation systems had to be modified to achieve better discernment in these cases.

Furthermore, data exploration is a desirable facility for this problem domain since the participants' responses are more distinctive in nature. The editors of a film project will generally have an idea of what is happening within a scene, so their work is more of a comparative nature than an explorative one. Therefore within the context of interview analysis, the key purpose of the timeline visualization changes from navigation of recording content to the exploration of it.

The primary focus of this application of the system is to align a list of questions to a recording of an interview. This allows researchers to quickly scan through the interviews without necessarily having to view them with a one-to-one time ratio. Instead of dividing up interview recordings between team members, the sections within every interview can be used. This makes it easier for teams of researchers to work on interview analysis while keeping a consistent analysis style. It is important to note here that the responses from the participants are not meant to be used as candidates for speech detection. Instead the idea here is to try to detect only the questions asked by the interviewer, as those rely mostly on the pre-defined question script. This discernment occurs during the speech detection phase itself using the question script.

While there were numerous minor modifications made to the analysis aspect of the software, one of the changes of note was the filter threshold for low confidence results. This threshold should be as high as 0.9 (up from around 0.5 for the digital film production application), due to the nature of the recordings. With digital film recordings, there is very little chatter in the footage; all dialogue is relevant to the script. However with user study interviews, only the interviewer's questions are considered for the speech detection. This means that any responses from the participants are technically considered to be noise by the speech detection engine. While this could be potentially problematic, using a higher threshold and relying on the sequence analysis algorithm for timeframe estimation gives results comparable to digital film production results as shown in section 7.1.5.

An additional feature provided for this problem domain is the ability to create bookmarks with interview files. During an interview key phrases can be spoken by the interviewers in order to bookmark specific times. This allows researchers analyzing the interviews to quickly locate specific points of interest of the interview without the interviewer having to write down the times in the recording where they occurred. The phrases themselves are user defined and must be specified within the question script file prior to analysis. For example, an interviewer could divide their interview into sections based on the type of questions being asked. A way to accomplish this is for the researcher to speak phrases such as “section one” at specific times during the interview. This could be used for dividing the interview recording in order to assign specific topics to different researchers for analysis or to allow one researcher to simply work on the topics out of chronological order.

7.1 Formative Evaluation

7.1.1 Methodology

This section describes a formative evaluation of the Interview Explorer system’s capabilities for aiding researchers in performing interview analysis tasks. The focus of this study is researchers who may not be experienced in analyzing Human-Computer

Interaction user studies or those who are not familiar with the specific interview content. If the system can be successfully utilized by this target, then the techniques of user study veterans should be improved by at least a small degree as well as they will have the best idea of how to take advantage of the functionality offered by Interview Explorer.

A total of five participants were asked to use the Interview Explorer software to examine a corpus of ten simulated interviews covering a small variety of topics. They were asked to complete a set of 15 tasks which ranged from navigating the audio content using the timeline visualization, to discovering trends between the recordings. The collected qualitative data from the evaluation was composed of both direct observation notes and feedback obtained through an interview session.

7.1.2 Participants

The population of this evaluation consisted of four students and one recent graduate from Dalhousie University. Each participant had at least minor experience with either running user studies or with analyzing the collected data. This level of experience is considered to be achieved as long as the participant had been a member of the research team of one or more user studies.

Three of the participants gained their experience through conducting and analyzing user studies during a graduate level course on research methods in human-computer interaction. One participant had gained experience conducting user studies for real world research projects. The last participant was included in the evaluation because of familiarity achieved through analyzing the results from a user study conducted in the workplace.

7.1.3 Interview Corpus used in the Study

Before the start of the evaluation, a corpus of ten simulated interviews was recorded to be analyzed by the participants. Each interview was based on a question script of approximately 75 questions of a variety of length. Some of these were short answer questions, while others were open-ended to generate lengthier responses to be more reflective of fact-finding discussions that would take place during an actual user study. Each audio recording featured a semi-structured interview, which means that some of question branches were skipped over if they were inapplicable.

Furthermore, additional questions were asked in some cases to probe for further information; similar to how a real interviewer would conduct the interview. On average, the interviews would last almost 20 minutes and the total set lasted over two and a half hours. These were all recorded using the built-in microphone of a laptop to investigate how the system functions with imperfect hardware. Furthermore, a variety of interviewees were enlisted as well to gauge the functionality of the software in close-to-real world situations. This semi-structured data set plays an important role in the evaluation of this software as it was previously explained that the somewhat unpredictable nature of these interviews could lead to poor speech recognition results.

7.1.4 Study Protocol

Each session in the evaluation began with a training session with the software after the participants consented to the study. In this phase of the evaluation, participants were given a guided introduction to the software's interface controls and general functionality. They were walked through example tasks that would be performed during the actual evaluation and any questions were answered about the software. Five to ten minutes of exploration time was then allocated in each session; giving each participant the opportunity for hands-on experimentation before any real task was assigned. At this point, participants were asked to complete 12 analysis-related tasks on the previously described interview corpus. Each of the assigned tasks fall into one of three general groups:

- Navigation to an area of the interview given a bookmark tag
- Finding the time in the recordings given a question from the interview script
- Discover any trends between interviews given a question or topic

Completion of each evaluation session took 40 minutes on average, and none of them were recorded. During each session a researcher was present to assign tasks one by one as well as to make direct observation notes based on the participants' reactions and interactions with the software. Occasionally some participants would stop to ask questions about the inner workings of the analysis software out of curiosity.

At the conclusion of every session, the participants were probed for additional feedback and impressions about the system. The questions from these post-session interviews are not included in this thesis as the interviews were generally unstructured and consistent of open-ended questions. The results of these post-evaluation interviews are discussed in the following section.

7.1.5 Results

7.1.5.1 Task Performance

Each participant in this evaluation was able to complete every one of the tasks they were assigned during the sessions. However, occasionally they would come across aspects of the software which would hinder their performance of task completion. For example, in some situations participants would find questions in the script which were not successfully recognized or estimated by the software. While this would generally happen no more than twice per interview recording, participants learned to quickly adapt to this situation through exploration. Questions from the script that were asked around the same time as the target question were selected from the script channel in order to find approximately where it occurs in the recording. These situations took participants no more than a few seconds to remedy.

Another issue that occurred during the evaluations was the inclusion of rare instances of imprecise time-frame predictions. In these situations, there would be as much as a 15 second offset from the correct position in the recording. While this would happen in three of the ten tasks assigned, by chance only two of the ten simulated interviews happened to be affected. Participants in the sessions dealt with this setback by simply listening to the recordings around the areas that the software estimated the questions were asked. While they were not familiar with the content of the simulated corpus, they were provided with a list of questions covered during the interviews which they would use to get their bearings during these situations. The time required to bypass this setback was relatively small overall; generally just taking up the amount of time equal to the size of the offset.

Arguably one of the biggest difficulties of the evaluation was the assigned tasks that required the participants to locate a specific question from the script channel. Since the user had no prior background knowledge of the simulated interview topics outside of skimming the list of questions, it involved aimlessly scrolling through the script channel. But we note that this problem has little to do with the actual function of the speech recognition and analysis software. Three out of the five participants agreed that there needed to be a textual search box for this situation. This is something that could easily be added with standard text searching methods.

7.1.5.2 Feedback Related to Functionality

The feedback related to the functionality of the Interview Explorer software showed that the participants were impressed with the system overall. After spending time with the software, participant P5 exclaimed that it was "so easy to find [a question in the recording]". This same participant also noted that while the software does not do everything for the user study analyst, the functionality that it does offer is like a "visual guide to an interview recording".

The time-frames being predicted during the evaluation were generally found to be within one question-answer pair. Each participant was able to successfully and efficiently locate their target questions without having to scan through the recordings manually.

Those taking part in the evaluation all agreed that they would rather use this system than listen to every minute in each recording. Participant (P1) noted that while the system was not perfect, it was definitely better than having to manually scan through the interview collection. One participant (P4) went as far as to ask for a copy of the software for an upcoming user study.

7.1.5.3 Feedback on Interface

While the functionality of Interview Explorer received positive overall feedback, the interface of the software ended up as being the primary area for future improvement. The agreement from all participants was that the timeline visualization needed to be more visible and more accessible. Making the boxes on the timeline taller along with a more visible highlighting on the selected sections was suggested by several participants.

Furthermore, the zoomable interface features were found to be confusing in general. One participant (P5) expressed a desire to see the script channel behaving in a way similar to Apple's iOS devices. This technique uses a momentum and friction system to manage the channel's scrolling velocity. To scroll through content, one has to merely flick a finger and the view slides in the desired direction.

In addition to this, four of the participants attempted to scroll through the script channel using the scroll wheel, which is meant for workspace navigation. This ended up being an area of frustration for the participants as they had a cognitive model of the panning controls that differed from the implemented model. One way to remedy this could be to use a context-sensitive approach to change the function of the scroll wheel depending on where the cursor is positioned. For example, while the cursor hovers over a script channel, scrolling the wheel would pan through the script content. However, using an identical action while the cursor hovers over the general workspace would have a different effect.

The last area of feedback from this evaluation dealt with the timeline visualization. Participant P1 suggested using lines to connect boxes from the script channel to their corresponding sections in the timeline visualization to improve the visual flow of the diagram. Aiming to accomplish this same end goal, participant P3 suggested that the colour scheme of the timeline should correspond with the questions in the script channel. For example, using a repeating pattern of colours or a colour gradient it could be easier to make connections between related data. On this note, some considerations need to be made to allow researchers with poor eyesight or colour-blindness to still be able to use the software.

7.2 Future Work for Interview Analysis

One of the primary limitations of the Interview Explorer system is its lacking support for semi-structured interviews. If the interview happens to feature questions which were asked out of order or it includes a high number of unwritten probing questions, the speech recognition systems will generally fail to produce acceptable results. While this system offers a bookmarking feature to help to reduce this from becoming a larger problem, it still requires a more comprehensive solution. One possible solution for this problem is to use a speech recognition system that is better able to detect spoken dialogue under a wider range of conditions and would not need to rely on a predefined interview script. This might also allow additional languages to be supported by Interview Explorer as well. However, one of the drawbacks of using such a system is that these speech engines are usually expensive.

Alternatively, it could be ideal to use speech recognition just to assign an order to the interview question list for semi-structured interviews. Allowing a pass of the speech recognition engine to rearrange the interview question list could potentially the sequence analysis algorithm could be used to its full extent without requiring additional user input. While this is already similar to how the system works as-is; the current software is not currently sensitive enough to achieve these results. Performing an extra pass of speech recognition without the interview script could be the first step in filling in this gap.

Aside from changes to the speech recognition systems, providing researchers with better tools for overriding the results within the software is a high priority future change. For example, allowing researchers to correct or modify the timeline while they are revisiting recorded sessions would be beneficial. This way the results could be incrementally improved through human input. Allowing metadata to be written on the timeline and the media channels would be favourable as well for managing large amounts of content. This would enable researchers at different locations to share their analysis notes without requiring separate documents.

Also in the future, a tool will be implemented to allow researchers to code individual responses during the analysis phase without requiring the interviews to be transcribed. The coding process involves tagging responses with a pre-defined series of categories and is performed to aid in the analysis of the interviews. The coding tool would work in conjunction with the timeline visualization tool so that each recognized section of the interview can be selected and tagged. The Axis Explorer visualization could then be customized to allow the interviews to be compared and evaluated using these coding categories using a custom plug-in. In this case, each axis would represent one or more coding categories. Furthermore, having the ability to output reports on the coding process would be an invaluable tool.

In addition to these future improvements, several modifications have been proposed based on the feedback from the evaluation. The first of these improvements aims to aid researchers in their search for specific questions in the script channel. A textual search box will be provided for this purpose. This feature could be used not only to help locate specific questions within the script channel, but also to locate questions coded under certain categories.

Several improvements to the timeline visualization are suggested here as well based on the evaluation feedback. One of the problems to be solved is the small size of the timeline sections. One solution to this would be to simply use a larger amount of screen space for the timeline. Smaller boxes would be more accessible this way unless there was a massive amount of recognized dialogue. An alternative solution to this would be to have one larger timeline which can be populated by the content of the others. This is a feature that was suggested by participant P4. This master timeline would be larger than the rest and provide better scrolling features. To populate the master timeline with one from the timeline channel, one would have to simply drag it onto master object.

The next area of improvement for the software is the timeline section highlighting feature. Participants found it to be difficult to see when there were many sections on the timeline. Even though the highlight colour was yellow and would stand out under normal circumstances, very narrow boxes seem to nullify that feature. One possible way around this is to increase the height of the timeline around the area of the highlighted boxes. By using a thicker timeline, a viewer would be able to better see where the highlighted boxes are in the timelines. Alternatively, using a fish eye lens over the timelines might yield good results.

One final future improvement for Interview Explorer is an upgrade which would allow research teams to better coordinate interview analysis endeavours. A networked software system is proposed here to aid in smoothing out the process of team-driven ventures. Project leaders would be presented with a team management interface which would allow them to assign workloads, facilitate contextual communication, and manage the results.

Workloads could be assigned by the project leader by simply highlighting areas of the workspace and tagging them with a team member's name. One of the highlights of this approach would be the options for the distribution. For example, instead of simply dividing up the raw interview files, areas of the question script could be segmented amongst the team. One of the advantages of this type of system is that individual team member would be searching relevant areas of the interviews for their specific topics instead of searching through large quantities of irrelevant files. Furthermore, by using a networked architecture, the interview recordings could be stored on a server and team members would be able to access pertinent files instead of having to sort through the whole database. However, this would also require the inclusion of sufficient security features for privacy reasons.

Communication between team members would be facilitated through a number of means. An in-software mailbox system would be used for direct communication between members. However, context-specific messages could be passed by making annotations directly onto the workspace. For example, voicing concerns over an aspect of one of the interview recordings could be accomplished by posting a note directly onto the corresponding timeline. This type of exact note posting system could help reduce miscommunications over which areas of the workload are affected by an issue.

Furthermore, the results of each researcher’s work could be quickly gathered using the software. One advantage of this idea is that the coding styles of different members could be analyzed. This could be important as some members may have a higher attention to detail than others. Establishing these differences could be key for assuring that the content has been analyzed sufficiently for the purposes of the research.

Chapter 8

Conclusion

The software system presented in this thesis provides a selection of tools for addressing both the digital film production and user study qualitative interview analysis problem domains. The first feature of the system is the pre-production media organization for digital film production provided by the zoomable workspace. This general interface provides users with a collection of widgets to help in organizing and navigating any media collected for the film. Media is represented by panels within the software which can be dragged around and organized. A variety of media can be represented by these panels, including images, video files, audio, and floor plans. Additionally, media panels can be connected to channel widgets for organizational purposes. These channels can be scrolled through and collapsed as the user requires for organization.

Several zooming and panning controls are also offered to users to help make larger workspaces more manageable. The first of these allows the user to maintain perceptual continuity while panning through the workspace at higher velocities. When users can rapidly scroll through the workspace, the camera view will zoom out to compensate. The second technique offers a way to zoom into specific areas of the workspace while allowing between control on the zoom speed and the focal point. The formative evaluation showed that for the Explorer views of our system, these navigation and zooming features tended to bog down the users and were not necessary.

Film scripts can be imported into the software where they each receive their own custom media channel. This specialized channel widget displays colour-coded information about which characters speak which lines of dialogue in addition to providing further interactive features. Selecting panels from script channels also highlights the corresponding sections from each of the timeline visualizations.

The speech recognition aspect of this software is used to analyze both the raw footage of film shoots and the recorded interviews of a user study. This will effectively locate the times in each video file when each line from the script is spoken. In the case of film production, this aligns the film's script to the footage. For user study interviews, the questions asked by the interviewer are aligned to the recordings of the interview to allow easier review.

While speech recognition is generally imperfect, the error removal algorithms presented in this thesis increase the speech engine's effectiveness. While filtering the raw speech recognition offers a moderate improvement to the results, a preferable approach was to analyze sequences of results and use them to validate individual results. Also, in addition to the removal of errors from the speech detection results, timeframes of dialogue elements that were not detected by the speech engine can be estimated by the software in an accurate fashion.

The results of aligning the script to the film footage can be visualized using timeline diagrams. These timelines show the detected timeframes of every detected and predicted line of dialogue in the script. A colour coding scheme shows the film editor which sections of the timeline were reliably detected and which were estimated by the system.

As previously mentioned, the sections on the timelines are connected to the script channel via an interactive layer. Selecting a section from a timeline not only highlights the corresponding line from the script, but also in each other timeline in the film bin. For user study interview analysis, this can allow researchers to find trends between interviews by focusing on the same questions in each recording. This interactive visualization for facilitating the exploration of digital film content and interview recordings is one of the primary contributions of this work.

Furthermore, the Axis Explorer visualization is a clip comparison tool used to visually inform the user of the degrees of similarity between multiple video clips simultaneously. This diagram provides users with an interactive way to explore their footage bins. While only a limited selection of plug-ins are presented in this thesis, additional axes can be created by the users via a programmable API to customize how they compare this data.

A short use case scenario was also presented to demonstrate how the system can be used in a real-world situation. A collection of video footage from a student film was analyzed using the tools provided by the Bin Explorer system. The script from this film was successfully aligned to the footage, allowing an editor to select specific areas of the footage quickly and accurately. In addition to this, the Axis Explorer visualization could allow an editor to customize their video sorting and comparison tools.

In addition to addressing the digital film production domain, this system has been modified to support interview analysis. Qualitative interviews are an important tool in human-computer interaction research as well as other fields. These come in three general variations: structured, unstructured and semi-structured. Interview Explorer aims to assist during the recording and analysis of both structured and semi-structured interviews, as these methods require the research teams to prepare lists of questions before conducting the interview.

Typically it can take a long time to fully analyze the data collected from qualitative interviews. Teams of researchers must review the notes and transcripts for each interview in the study while sometimes having to refer to the original interview recordings for clarification or review. Depending on the quantity of participants involved in the study, on occasion the interview data will have to be divided up between members of the research. While this can save time, it can also mean that different analysis styles will be used on the data sets leading to heterogeneous results being synthesized. One of the benefits Interview Explorer offers is an alternative distribution option to assigning work. Instead of dividing individual interviews for separate analysis, the content itself can be divided by aligning the question script to the recordings. This allows members or subgroups to focus on topical areas of all interviews in the study instead of multiple themes in a small set. This approach to supporting interview analysis is one of the contributions offered by this work.

Furthermore, a formative evaluation of the software system is offered to show how it works in situations with real users. Participants in the evaluation were asked to analyze a corpus of simulated interviews in a similar fashion to how a study analysis would work. The direct observation notes and feedback from the participants show that they found the software’s data exploration capabilities and utility powerful and simple to use. While the feedback from the users showed that the interface had room for improvement, the users were undivided in their agreement that the Interview Explorer system offered a new and important technique for interview recording analysis.

In summary, both the Bin Explorer system and the Interview Explorer variant have met success for their respective problem domains. Case study usage of the Bin Explorer system for analyzing film footage has been met with success; the error removal and estimation methods were able to drastically improve the speech recognition to allow exploration of the video content. Furthermore, a formative evaluation of the software for qualitative interview analysis assistance has provided promising results. The participants were all able to complete their assigned tasks and felt that Interview Explorer offered a valuable tool for interview exploration.

Bibliography

[1] Seigel, J., Fisher, S. and Brooks, S., "Towards a Unified System for Digital Film Production." Vancouver, Canada : Springer LNCS, October, 2011. In Proceedings of the 10th International Conference on Entertainment Computing. pp. 149-154.

[2] Final Draft, Inc., Final Draft. *http://www.finaldraft.com/.* [Online]

[3] Spence, R., *Information Visualization.* s.l. : ACM Press, 2001, pp. 85-88.

[4] Igarashi, T. and Hinckley, K., "Speed-dependent automatic zooming for browsing large documents." San Diego, CA : ACM, 2000. Symposium on User Interface Software and Technology. pp. 139-148.

[5] Cockburn, A. and Savage, J., "Comparing speed-dependent automatic zooming with traditional scroll, pan and zoom methods." Bath, England : University of Canterbury, 2003. People and Computers XVII: British Computer Society Conference on Human Computer Interaction. pp. 87-102.

[6] Malacria, S., Lecolinet, E. and Guiard, Y., "Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: the cyclostar approach." New York, NY : ACM, 2010. In Proceedings of the International Conference on Human Factors in Computer Systems. pp. 2615-2624.

[7] Perlin, K. and Fox, D., "Pad: an alternative approach to the computer interface." New York, NY : ACM, 1993. In Proceedings of Computer graphics and interactive techniques. pp. 57-64.

[8] Wardrip-Fruin, N., Meyer, J., Perlin, K., Bederson, B. and Hollan, J., "A zooming sketchpad, a multiscale narrative: Pad++, PadDraw, Gray Matters." New York, NY : ACM, 1997. In ACM SIGGRAPH 97 Visual Proceedings: The art and interdisciplinary programs of SIGGRAPH '97. p. 141.

[9] Bederson, B., Meyer, J. and Good, L., "Jazz: An extensible zoomable user interface graphics toolkit in Java." New York, NY : ACM, 2000. In Proceedings of User interface software and technology. pp. 171-180.

[10] Bederson, B., Grosjean, J. and Meyer, J., "Toolkit Design for Interactive Structured Graphics." 2004. IEEE Transactions on Software Engineering. Vol. 30(8), pp. 535-546.

[11] Haller, H. and Abecker, A., "iMapping: a zooming user interface approach for personal and semantic knowledge management." New York, NY : ACM, 2010. In Proceedings of Hypertext and Hypermedia. pp. 119-128.

[12] Greyfirst Corporation, Celtx. *http://www.celtx.com.* [Online]

[13] Adobe, Story. *http://www.adobe.com/products/story.* [Online]

[14] Yi, J. S., Melton, R., Stasko, J. and Jacko, J., "Dust & Magnet: multivariate information visualization using a magnet metaphor." Minneapolis, MN : Information Visualization, 2005, Vol. 4, pp. 239-256.

[15] Moustafa, R. and Wegman, E., "On Some Generalization to Parallel Coordinate Plot." *A Data Visualization Workshop.* Rain am Lech (nr.), Germany, 2002.

[16] Chambers, J. M., Cleveland, W. S., Kleiner, B. and Tukey, P. A., "Graphical Methods for Data Analysis." Boston : Wadsworth International, Duxbury Press, 1983.

[17] Lazar, J., Feng, J. and Hochheiser, H., *Research Methods in Human-Computer Interaction.* London : Wiley, 2010.

[18] Qualrus, *http://www.ideaworks.com/qualrus/.* [Online] http://www.ideaworks.com/qualrus/.

[19] Atlas.ti, *http://www.atlasti.com/.* [Online]

[20] Hall, M. and Brown, B., TagPad. *http://itunes.apple.com/ca/app/tagpad/id443465566.* [Online]

[21] Dropbox Inc., Dropbox. *www.dropbox.com/.* [Online]

[22] Li, Y., Zhang, T. and Tretter, D., "An overview of video abstraction techniques." *Tech. Report HPL-2001-191.* 2001.

[23] Lienhart, R., Pfeiffer, S. and Effelsberg, W., "Video abstracting." *Communications of the ACM.* 1997, Vol. 40, 12, pp. 55-62.

[24] Eickeler, S., Wallhoff, F., Iurgel, U. and Rigoll, G., "Content-based indexing of images and video using face detection and recognition methods." Washington, DC : IEEE Computer Society, 2001. In Proceedings of the Acoustics, Speech, and Signal Processing. Vol. 3, pp. 1505-1508.

[25] Smith, M. and Kanade, T., "Video skimming for quick browsing based on audio and image characterization." *Carnegie Mellon University, Tech. Report CMU-CS-95-186.* 1995.

[26] Wolf, C., Jolion, J. and Chassaing, F., "Text localization, enhancement and binarization in multimedia documents." Quebec, Canada : *ICPR, 4.* 2002, pp. 1037-1040.

[27] AutoDesk, AutoCAD. *http://usa.autodesk.com/autocad/.* [Online]

[28] Agarawala, A. and Balakrishnan, R., "Keepin' it real: pushing the desktop metaphor with physics, piles and the pen." New York, NY : ACM, 2006. SIGCHI. pp. 1283-1292.

[29] VideoLAN, *VLC.* [Online] http://www.videolan.org/vlc/.

[30] eTeks, SweetHome3D. *http://www.sweethome3d.com.* [Online]

[31] Microsoft, *Microsoft. Speech API.* [Online] http://www.microsoft.com/speech.