Navigating Video Using Structured Text

Amy Pavel

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2019-78
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-78.html

May 17, 2019

Navigating Video Using Structured Text

by

Amy Pavel

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Björn Hartmann, Co-chair
Professor Maneesh Agrawala, Co-chair
Professor Eric Paulos
Professor Abigail De Kosnik

Spring 2019

Navigating Video Using Structured Text

Abstract

Navigating Video Using Structured Text

by

Amy Pavel

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Björn Hartmann, Co-chair

Professor Maneesh Agrawala, Co-chair

Video has become a primary medium for producing and consuming entertainment, educational content, and documented events. As the cost of recording, sharing, and storing video has decreased, the prevalence of video has increased. Yet, video remains challenging to use as an informative medium because it is difficult to search, browse and skim the underlying content.

Using a timeline-based video player, users must scrub back-and-forth through a video to gain an overview or locate content of interest. Some video viewing interfaces allow users to search and browse videos by text transcripts or caption files. However, transcribed and captioned speeches or conversations often contain disfluencies and redundancies typical of speech which make the transcripts time-consuming to read and difficult to skim. Further, transcripts and captions lack structure – transcripts consist of long blocks of text, while captions have a series of short phrases. Without structured organization, it can be difficult for viewers to browse topics or get a high-level overview of the content.

This thesis explores new ways to search, browse and skim videos through structured text. We aim to create navigable representations of videos that let users ask questions of the video as efficiently and flexibly as possible, and facilitate low cost production. This thesis introduces systems that embody these goals, using structured-text documents aligned to video to enable efficient and flexible video searching, browsing and skimming across three domains: (1) informational lecture videos, (2) films, and (3) casually recorded feedback.

To Mom, Dad, Zach, and Steve.

Contents

List of Figures

List of Tables

Acknowledgments

First, many, many thanks to my advisors Maneesh Agrawala and Björn Hartmann for their constant mentorship and support starting in my undergraduate years and continuing through the end of my graduate degree. It is difficult to summarize the impact that they've had on my work, and I feel tremendously lucky to have received their guidance.

Thanks additionally to the members of my thesis committee, Gail De Kosnik and Eric Paulos, for providing feedback from new perspectives and helping me cross the finish line. I've also collaborated with a number of people throughout my time at Berkeley and my research would not have been possible without them. Particular thanks to my first research role model Floraine Berthouzoz, my internship mentor at Adobe Dan Goldman, Berkeley graduate students Colorado Reed, Jeremy Warner, Wei Wu, and Berkeley undergraduate students Tonya Nguyen and Kaushik Kasi.

Through many years at Berkeley, my friends and lab mates have provided me with an endless supply of encouragement, advice, and fun. Thanks to Alex Hall and Eric Yao for helping me hold down the fort of Maneesh's lab North in 523 Soda. Thanks also to the 523 Soda residents before them for the lively conversations, deadline support network, and life/research advice in my first few years: Colorado Reed, Floraine Berthouzoz, Jonathan Harper, Sean Arietta, Kenrick Kin, Robin Held, and Jessica Hullman among others. My informal office mates and fellow Floraine mentees Anh Troung and Jane E helped me feel at home in Maneesh's lab South. Thanks to the first b-crew, Valkyrie Savage, Peggy Chi and Shiry Ginosaur for warmly welcoming me into the group as an undergraduate, and the more recent b-crew including Andrew Head, Jeremy Warner, Eldon Schoop, Jingyi Li, Elena Glassman, Will McGrath, James Smith, and Bala Kumaravel and many more for invaluable project feedback, and forming a supportive research community. A few extra thanks to Cesar Torres for being a reliable grad school companion, to Nate Weinmann for all of the CS 160 chats and his excellent course support, to Philippe Laban for the many coffee and lunch breaks, and to Rachel Albert for her friendship and outdoor adventures.

Thanks also to everyone I met before this whole grad school thing. I was lucky enough to live with a bunch of great friends who provided built-in socializing during crazy deadline pushes, and encouraged me to do lots of fun things outside of those times. I'm happy that I got to experience so much of what the Bay Area had to offer with all of them. Thanks to the 5th floor Berkeley crew for unending positivity and encouraging me to celebrate every now and then.

Finally, to my long-time friends, to my lab mate turned partner Steve Rubin, and to my family, Valerie, Stephen and Zach Pavel – thanks for everything.

Chapter 1

Introduction

Video is a primary medium for producing and consuming entertainment (e.g., movies, YouTube vlogs, sports), educational content (e.g., recorded lectures and seminars, tutorials), and documentation (e.g., home movies, newsworthy events). Video enables access to otherwise inaccessible events, and the combination of motion and sound creates uniquely engaging content. As the cost of recording, sharing, and storing video has decreased, the prevalence of video has increased – now accounting for at least 75% of all IP traffic [41]. Yet, video remains challenging to use as an informative medium because it is difficult to search, browse and skim.

The traditional video player included in web browsers [52, 138, 145], editing tools [110, 53], and personal playback applications [114, 19, 111] features the ability to play/pause a video and a scrub, or navigate, the video timeline. The timeline lets users preview the frame-by-frame visual content while navigating the video, but the timeline does not reveal audio or higher level content (e.g., story elements, topics). To gain a high-level overview of the video or locate content of interest for specific tasks, users need to scrub back and forth through the timeline, typically playing and pausing the video multiple times. Some video viewing interfaces allow users to search and browse videos by text transcripts [15, 27, 54, 86] or caption files [35, 81]. Such transcriptions support keyword search and navigation based on speech. However, disfluencies [70] and redundancies typical of speech make these transcripts time-consuming to read and difficult to skim. In many domains, such low-level navigation mediums (e.g., frame-by-frame timeline navigation, or transcribed text navigation) do not support the high-level searching, browsing, and skimming tasks that users want to complete with videos (e.g., figuring out what topics are covered in a video, rewatching a particular scene or subject matter segment, or recalling the main feedback you were given in a recorded design conversation).

This thesis explores new ways to search, browse and skim videos through structured text, leveraging users' existing strategies for using text as an informative medium. We aim to create navigable representations of videos that let users ask questions of the video as efficiently and flexibly as possible, and facilitate low cost production. This thesis introduces systems that embody these goals, using structured-text documents aligned to video to enable efficient and flexible video searching, browsing and skimming across three domains: (1) informational lecture videos, (2) films, and (3) casually recorded feedback.

1.1 Challenges

We aim for our video representations to 1) allow efficient navigation within the video, 2) flexibly support navigation for existing and new domain tasks, and 3) afford low cost generation (e.g., time, effort).

Efficiency

To achieve efficient high-level searching, browsing, and skimming tasks, we use structured text. While lower-level representations may be more appropriate for lower-level video manipulation tasks, such a higher-level representation can leverage strategies for searching, browsing, and skimming in text. Long-standing work shows that specifically structured organization of text supports locating main ideas and facilitating comprehension [108]. We use such prior work in text processing to inform design guidelines for our structured text representations. For instance, efficient representations will match the conceptual structure of the content, and when possible leverage a user's prior knowledge of general and domain-specific text structure [108]. Thus, we take a domain-specific approach to crafting our structured text navigation tools (Table 1.1). In Chapter 2, we further discuss structure in text and video, and how we leverage research on structured text to support our system designs.

Flexibility

We aim to let users flexibly complete existing and new tasks within each given domain. To achieve such flexibility, we 1) understand the set of existing tasks within the domain of interest using interviews and literature search, and 2) provide multiple text structures that vary in level of abstraction and/or order of information (Table 1.1). Through providing multiple versions of structured text, we can support different conceptual structures (e.g., plot points vs shots that illustrate that plot point, high-level topics vs a teacher's quoted definition). We discuss the domain and task considerations, and the corresponding designs of multiple structured text representations in Chapters 4-6.

Low cost

Ideally, an expert might generate the structured text and align the text to the content of interest in the video (e.g., Bret Victor's hand-crafted video digest [136]). Although such expert-generation may yield high-quality navigational tools, the process of creating such a tool is time intensive. Instead, according to our third goal, we seek hybrid automated (e.g., natural language processing, vision), crowd, and expert methods that facilitate efficient generation of navigable structured text documents (Table 1.1). We discuss our selected methods and relevant prior work throughout our system designs in Chapters 4-6. In Chapter 7 we discuss future work, including promising directions for automating additional aspects of our systems.

	Efficient representation		Flexible navigation		Low-cost generation	
	Domain	Structured text	Levels of abstraction	Ordering	Segmentation	Abstraction
Video Digests	Informational videos	Textbooks Outlines	Chapter titles Section summaries	Chronological	Automatic Expert	Crowd Expert
SceneSkim	Film	Captions Scripts Summaries	Captions Scripts Summaries	Chronological	Automatic	Existing
VidCrit	Feedback sessions	Edit lists	Transcribed edit User-revised edit	Chronological Shot-based	Automatic	Expert

Table 1.1: Our approaches for efficient representation include focusing on a domain such that we can rely on prior knowledge of text. We achieve flexible navigation through providing different representations both through levels of abstraction and ordering of the indexable text. Finally, we create low cost approaches for generating such navigational tools using hybrid expert, crowd, and automatic systems.

1.2 Contributions

This thesis explores using structured-text documents aligned to video to enable efficient and flexible search, navigation and skimming of video content. We aim for a world in which it is as easy to use video as a reference and research tool as it is to use written text documents. We have created 3 prototypes meant to investigate different parts of the design space of structured text documents for navigation. Towards this aim, this thesis makes the following contributions:

- This thesis introduces a method for designing and creating structured text representations of video exemplified through three systems. For each domain we 1) investigate video-based tasks within the domain, 2) find relevant examples of structured text in that domain, and 3) design hybrid automatic, expert and/or crowdwork methods to create indexes into video based on the structured text. To achieve efficient representations, we propose design guidelines for structured text navigation tools based on prior work in reading science (Chapter 2).

- An approach for authoring video digests (Chapter 4), a format inspired by Bret Victor's indexable seminar video [136] for searching, browsing and skimming informational lecture videos. Mimicking a textbook or lecture outline, a video digest segments the video into navigable chapter and section segments, and provides short text summaries and thumbnails to surface content in each segment. Our approach lets authors create manual, automatic, or hybrid manual-automatic video digests through providing: 1) a transcript-based interface for segmentation and summarization, 2) an automatic hierarchical segmentation approach, and 3) a crowdsourced summarization pipeline. Our evaluation (n=192) suggests that manually and automatically created video digests help users produce summaries containing more important points when given a short amount of time to view the video. We also find our hybrid manual-automatic approach helps authors create digests more efficiently.

- SceneSkim (Chapter 5), a system built to support searching, browsing, and skimming within film. For many films — unlike for most informational lecture videos — there already exist

structured text documents that summarize the video content (e.g., captions, scripts, and plot summaries). SceneSkim contributes 1) an interface for searching, browsing and skimming within films based on existing text documents, and 2) algorithms for producing alignment between the original film, captions, scripts, and plot summaries. Our evaluation suggests SceneSkim helps efficiently answer existing questions from film studies literature, and novel questions from film practitioners.

- VidCrit (Chapter 6), a tool that lets video reviewers record their feedback and allows video editors to easily skim and browse spoken feedback using topic-segmented text comments. While video reviewers may not normally record their spoken feedback using video, our evaluation suggests that, given a fixed amount of time, video reviewers record more feedback when speaking their critiques rather than writing their critiques. VidCrit additionally contributes 1) an interface for recording feedback, 2) an interface for reviewing recorded feedback, and 3) algorithms for segmenting and labeling feedback text. Our evaluation with video reviewer/video editor pairs suggest the topically segmented comments help the reviewer to process feedback efficiently.

1.3 Overview

In the Chapters 2-3 of this thesis, we give a brief background on structure in text and video, propose design guidelines for creating structured text, and discuss related work in the area of video interaction. The following three chapters discuss each prototype in turn. Finally, the conclusion restates the contributions and outlines future work. This overview provides summaries of each chapter:

Chapter 2: Structure in text and video

As many low-level video interfaces do not suit higher level searching, browsing and skimming tasks, we use structured text. We discuss properties that make structured text a preferable medium for indexing videos and suggest design guidelines for creating structured text (according to prior work in reading science). We include a brief discussion on existing automatic methods for synthesizing structure in unstructured text. Finally, we define terms related to structured information about video produced before, during and after video production.

Chapter 3: Related work

In Chapter 3, we discuss video interaction techniques in domains including video annotation, video summarization, video navigation, and video editing. Within video navigation, we discuss prior work that covers navigation based on video structure including keyframes and transcripts. In addition, we include work that specifically addresses using domain-specific metadata to navigate within video (e.g., sports statistics to navigate baseball clips). Our work draws from algorithms and navigation techniques in prior work but addresses novel domains and question types through structured-text summaries.

Chapter 4: Video digests

Video digests are a structured text format for informational videos that afford browsing and skimming by segmenting videos into a chapter/section structure and providing short text summaries and thumbnails for each section. Viewers can navigate by reading the summaries and clicking on sections to access the corresponding point in the video. Chapter 4 presents a set of tools to help authors create such digests using transcript-based interactions. With our tools, authors can manually create a video digest from scratch, or they can automatically generate a digest by applying a combination of algorithmic and crowdsourcing techniques and then manually refine it as needed. Feedback from first-time users suggests that our transcript-based authoring tools and automated techniques greatly facilitate video digest creation. In an evaluative crowdsourced study we find that given a short viewing time, video digests support browsing and skimming better than timeline-based or transcript-based video players.

Chapter 5: SceneSkim

Searching for scenes in movies is a time-consuming but crucial task for film studies scholars, film professionals, and new media artists. Our formative interviews reveal that such users search for a wide variety of entities — actions, props, dialogue phrases, character performances, locations — and they return to particular scenes they have seen in the past. Today, these users find relevant clips by watching the entire movie, scrubbing the video timeline, or navigating with opaque DVD chapter menus. Chapter 5introduces SceneSkim, a tool for searching and browsing movies using synchronized captions, scripts and plot summaries. Our interface integrates information from different documents to allow expressive search at several levels of granularity: Captions provide access to accurate dialogue, scripts describe shot-by-shot actions and settings, and plot summaries contain high-level event descriptions. We propose new algorithms for finding word-level caption to script alignments, parsing text scripts, and aligning plot summaries to scripts. Film studies graduate students evaluating SceneSkim expressed enthusiasm about the usability of the proposed system for their research and teaching.

Chapter 6: VidCrit

Video production is a collaborative process in which stakeholders regularly review drafts of the edited video to indicate problems and offer suggestions for improvement. Although practitioners prefer in-person feedback, most reviews are conducted asynchronously via email due to scheduling and location constraints. The use of this impoverished medium is challenging for both providers and consumers of feedback. Chapter 6 introduces VidCrit, a system for providing asynchronous feedback on drafts of edited video that incorporates favorable qualities of an in-person review. This system consists of two separate interfaces: (1) A feedback recording interface captures reviewers' spoken comments, mouse interactions, hand gestures and other physical reactions. (2) A feedback viewing interface transcribes and segments the recorded review into topical comments so that the video author can browse the review by either text or timelines. Our system features novel methods to automatically segment a long review session into topical text comments, and to label such

comments with additional contextual information. We interviewed practitioners to inform a set of design guidelines for giving and receiving feedback, and based our system's design on these guidelines. Video reviewers using our system preferred our feedback recording interface over email for providing feedback due to the reduction in time and effort. In a fixed amount of time, reviewers provided 10.9 (σ=5.09) more local comments than when using text. All video authors rated our feedback viewing interface preferable to receiving feedback via e-mail.

1.4 Statement of Multiple Authorship and Prior Publication

This thesis is based on the following previously published papers: Video Digests at UIST 2014 [106], SceneSkim at UIST 2015 [103], and VidCrit at UIST 2016 [104]. I am the primary author on each publication and I led all of the corresponding projects. But, this research could not have been completed without my advisors Maneesh Agrawala and Bjoern Hartmann and my coauthors who I've been lucky to work with. In particular, Dr. Dan Goldman, my Adobe internship mentor, provided valuable ideas and advice for two projects (SceneSkim [103], and VidCrit [104]). Colorado Reed also contributed to the design, algorithm selection, and implementation of Video Digests [106].

Chapter 2

Structure in text and videos

In this chapter we investigate relevant types of structure in text and video, and their benefits and outline the approach used for systems in the rest of the thesis.

2.1 Structure in text

We select structured text documents aligned to video as an efficient method for higher level searching, browsing and skimming tasks. Text lets users leverage their existing knowledge and tools for searching, browsing and skimming text documents while reading. Longstanding research in reading science suggests that structural aspects of text can improve cognitive processes such as attention, reading, comprehension, memory and search [83]. We leverage such prior work to understand the advantages of structure in text, and how we can use such advantages to create video navigation tools.

What is structure in text?

First, what do we mean by structure? We use the term structure to refer to the organization of text rather than the content of the text. Surface structure features used for organization include ordering, levels of abstraction, explicit linguistic cues, and typographic cues. Such forms of structure in text give us a set of tools for expressing structure in our text representations in video. We define each type of structure in turn:

Ordering: The order of words in a sentence and sentences within the document can guide the reader by suggesting importance and relationships. For instance, the first sentence of a paragraph may be interpreted as the topic sentence, impacting what users find to be the most important idea [126].

Levels of abstraction: Expository text often provides previews of content at multiple levels of abstraction. For instance, titles, headings and subheadings (often indicated with font and spacing) help designate themes of different text segments which can help readers better search for specific

information [64]. In addition, high-level overviews and summaries of text (e.g., an abstract) can help readers interpret the structure of a document, and identify main ideas.

Explicit linguistic cues: Words and phrases in the text itself can be used to signify structure, to show relationships between content, or to highlight important ideas. Examples of explicit linguistic cues include: enumeration cues (e.g., Second, Next), connective cues (e.g., because, however), abstraction cues (e.g., In summary), and ordering cues (e.g., Recall, Later).

Typographic cues: Typographical cues can be used redundantly with other types of signaling (e.g,. headings often designated with different spacing and font), but they can also be used independently to signal important information[83] such as new terms to be defined.

Exploiting the benefits of structure

While the presence of such organizational features can independently impact processing of text (i.e. text-driven processing), the impact may be strengthened by mirroring the conceptual structure of the content, and by interacting with a reader's prior knowledge of domain structure (i.e. knowledge driven processing). The survey by Goldman and Rakeshaw, "Structural Aspects of Constructing Meaning" [59], reviews prior research to draw conclusions about the impact of structure on text-driven processing and knowledge driven processing. For a complete list of conclusions and relevant study findings see the original paper. We summarize the conclusions most relevant to this thesis below:

Conclusion 1 Structural cues (e.g., order, cueing) can improve identification and memorability of main ideas.

Conclusion 2 Parallelisms between surface structure text and the underlying conceptual structure of the information facilitate comprehension – and comprehension is hampered when the two are not aligned.

Conclusion 3 Making the structure of the text more salient improves comprehension.

Conclusion 4 Readers use their knowledge of structure in processing text.

Conclusion 5 Knowledge of structural forms of text develops with experiences with different genre.

Such conclusions suggest methods for making design more usable through creating a salient structure, and making sure that there is a match between the given text structure and the users' prior knowledge of text structure. Using these conclusions drawn from prior research, we propose design guidelines to support efficient video searching, browsing and skimming using structured text documents:

- **Structure saliency** The selected structure of text should be salient (Conclusions #1 and #3)
- **Structure-concept match** The structured text should mirror the conceptual content in the video relevant to searching, browsing and skimming tasks within the given domain (Conclusion #2).
- **Structure knowledge** Structured text representations should follow typical structure conventions, and when possible, they should mirror the structure of text encountered in the domain of interest (Conclusions # 4 and #5).

The guidelines of structure-concept match and using structure knowledge share similarities with the Norman's design goal of decreasing the gap between the interface designer's mental model and the users' mental model of the interface's operation [100]. In design, Norman suggests that mental models may be based on mappings, affordances and constraints which themselves can be influenced by a user's prior knowledge. In this thesis, we use existing knowledge of how application of such prior concepts can impact user performance.

In order to meet such design guidelines, we seek to make the structure of the text salient by relying on the forms of structure proposed above (e.g., ordering, levels of abstraction, linguistic cues and typographic cues). However, we decide what types to use based on the appropriateness for the conceptual content in the video, and in the domain of interest. In addition, we consider the cost and technical practicality of generating the structural form. For instance, it may be challenging to generate summaries or overviews, but it is straight forward to apply a different ordering given a set of tags.

Extracting structure in text

Given a piece of continuous text (i.e., the transcript of a video as described below), could we automatically generate our structured text for our navigation tools? In this section, we cover a few relevant automatic text-processing methods. Specifically, we discuss automatic approaches for text summarization. In the remainder of this thesis, we use hybrid automatic, crowd, and expert approaches to address weaknesses in the automatic methods alone.

Text summarization

The task of text summarization involves condensing a piece of text into a shorter representation of the main points. Text summarization can be divided into two alternative approaches [98]: *extractive* summaries concatenate existing text fragments from the document, while *abstractive* summaries generate new language to convey the main topics of the text. As video transcripts often contain disfluencies and redundancies typical of speech, directly concatenating portions of the transcript into an extractive summary often generates incoherent results. Therefore, we focus on primarily abstractive summaries for video transcripts where possible. But, because current algorithmic techniques can not yet produce human quality abstractive summaries [63], despite recent advances, in this thesis we use other approaches (e.g., crowdsourcing, expert generation) to generate them.

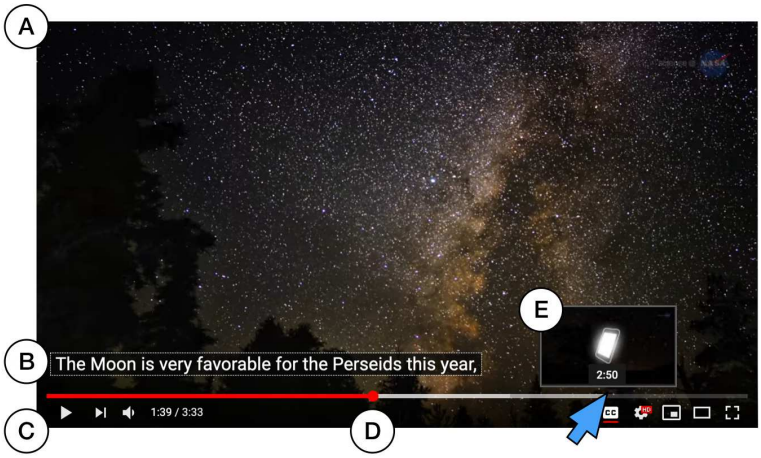

Figure 2.1: YouTube's video player displaying the video. This video player, similar to other modern video players, features (A) the main video screen playing at the current time, (B) corresponding closed caption of the speech, (C) play/pause button with navigable timeline that displays (D) current progress in red, and (E) a thumbnail preview of later video clip as cursor hovers over the timeline. Video source: NASA ScienceCasts: Nature's Fireworks Show in August from ScienceAtNASA, licensed under CC BY 3.0.

Text segmentation

The task of text segmentation consists of breaking a piece of text into topical segments. For instance, paragraph boundaries represent a text segmentation roughly based on topical boundaries. However, prior work from Stark et al. suggests that when asked to generate such text segments in unsegmented text, humans produce varied results [126] even though the people segmenting the text express the same reasoning (that they segmented based on topic changes). As a number of topical segmentations may be considered correct, and segmentation does not require machine generation of text, we can apply automatic topical segmentation when relevant (Chapter 4, Chapter 6). We use a popular method that optimizes for lexical coherency [50] for Chapter 4, and create a domain-specific segmentation method for video critiques (Chapter 6).

2.2 Structure in video

Our goal is to create or align structured text representations for recorded videos. The aligned structured text representation mirror the conceptual structure of content in the video (specifically, related to questions of interest). To do this, we first need to understand types of structured information that exists alongside videos. First, we define standard terms referring to structure within recorded video footage:

Video consists of moving visual images, often accompanied by audio such that multimedia formats typically encompass both (e.g., MP4 and WebM). We will refer to the moving visual im-

ages along with the audio component together as *video*. Typically, users view video using a video player which features a play/pause button and a navigable timeline (Figure 2.1).

Frame A video frame is a static image (i.e. rectangular raster of pixels). Videos consist of many frames played back at a consistent rate (e.g., 24 frames per second). Frames, or keyframes (frames selected to represent a section of content, e.g., thumbnails selected every 10s) remain the main unit of navigation for video content (typically presented to user as the time in the video).

Shot A shot is a continuous set of footage between two cuts or edits. In filming, it may be considered a single continuous filming segment. A video with no editing or cuts consists of one shot. An edit decision list (EDL) may contain shot boundaries from editing, or they can be detected after the fact in an edited video using shot detection (using features like color histograms, optical flow, or edge detection [123]).

Beyond frames and shots contained within the video footage, there exist documents created before (during the planning phase) and after (during the presentation stage) the filming and editing stages of the video are finished. We start with low-level documents created after the video is complete, and then finish with higher level documents completed before filming:

Subtitles (captions) Subtitles (Figure 2.1B) typically represent a transcription of the speech in the video or a translated transcription of the speech in the video, played alongside the video content in 3-5 word chunks. Subtitles may also contain information about the audio aside from the speech content including sound effects, music, and the name of the speaking character. Captions may be provided by the studio or created after the fact using automatic [60] or crowd-powered [11] techniques.

Aligned transcripts Similar to subtitles, some videos feature an aligned transcript especially for educational lecture videos. An aligned transcript, like subtitles, is a transcription of the speech included in the video. The transcript is then aligned to the video such that clicking a line in the transcript will play back the line in the video. Similar methods can be used to obtain transcripts as to obtain captions. A word-level alignment (rather than phrase-level) between the words and the video may be obtained through forced-alignment between word phonemes and audio features during post-processing [144, 101].

Titles and summaries A video title, synopsis, or plot summary (e.g., IMDB, Wikipedia) contain themes or important content included in the video. Video titles and short synopsis of the content may be produced before or after video production. Longer plot summaries available online are generated after the release of the video.

Script Here, we refer to a script as a text document used to plan the video content. Depending on the type of video and the intended audience of the script, it may take different forms. For instance, a camera person or an editor may receive a shot list, or shooting script (Figure 2.2). A person producing a film, or an actor may receive a screenplay (Figure 2.3, as used in Chapter 5).

Shot #	Duration	Shot type	Motion	Description
1	0:00-0:03	CU	Static	Close up on girl's face looking excited
2	0:03-0:05	MLS	Dolly	Family walks towards recognizable theme park
3	0:05-0:07	CU	Dolly	Girl walking notices something exciting
4	0:07-0:09	CU	Static	Amusement park ride moving

Figure 2.2: A shooting script, or a shot list, may be used to plan footage to shoot in a video. This partial shooting script for a theme park advertisement includes the shot number, the shot type, or distance of the camera to the subject, expressed in acronyms (e.g., CU is close up, MLS is medium long shot), camera motion, and a shot description. The shot list may also include attributes like camera angle and type of audio that will back the track.

```
EXT. JOHN AND MARY'S HOUSE - CONTINUOUS

An old car pulls up to the curb and a few KNOCKS as the
engine shuts down.

MIKE steps out of the car and walks up to the front door. He
rings the doorbell.

                                                    BACK TO:


INT. KITCHEN - CONTINUOUS

                    JOHN
          Who on Earth could that be?

                    MARY
          I'll go and see.

Mary gets up and walks out.

The front door lock CLICKS and door CREAKS a little as it's
opened.
```

Figure 2.3: The screenplay features elements such as interior (INT.) and exterior (EXT.) setting names, descriptions of actions in the scenes, and character dialogue. We discuss scripts in more detail in Chapter 5. Script source: Script by Wikipedia user Mendaliv under public domain.

Casual scripts also exist in the case of other recorded events. For instance, a person recording YouTube video might use an outline of the topics to cover, and a seminar speaker may use presenter notes.

Chapter 3

Related work

Schoeffmann et al.'s survey of video interaction work suggests 7 main areas of video interaction work: video annotation, browsing, editing, recommendation, navigation, retrieval, and summarization [120]. As we focus on navigation within a single long video, the areas of video annotation, browsing, editing, navigation and summarization relate most closely. We summarize a subset of work most related to this thesis in these areas. As we also consider how to condense existing text representations of video, we also consider methods for creating structure from existing text (e.g., paragraph segmentation, automatic summarization). We additionally discuss work specifically related to each system (and supporting algorithms) in the system descriptions (Chapter 3-5).

3.1 Video annotation tools

Video annotation tools allow users to manually watch and label a video (e.g., semantic labels or segments) according to its content. Annotation tools rely on video navigation techniques such that they might help users accurately select annotation targets [120]. In some cases, labels created by video annotation techniques could be used to create navigation tools based on those labels. Early systems for logging videos addressed individually logging clips during the recording session [85], after the recording session [140], or as a group after the recording session [42, 43, 18]. Later work improved video logging usability and efficiency by creating fluid tablet interactions [113], allowing users to apply tags to keyframes [139], accommodating voice annotations [132], and segmenting videos at the per frame level [47].

Whereas previous work mainly addresses generalized tasks of assigning tags and writing notes on video clips (e.g., for editing, behavioural video coding, and analysis), we consider creating specialized annotations for the purpose of later searching, browsing and skimming. Each project uses a new approach for assigning video annotations. Video Digests (Chapter 4) allows authors to construct hierarchical chapter and section summaries using the transcript. SceneSkim (Chapter 5) automatically applies annotations to video using alignment to existing text documents (therefore, we do not require explicit video annotation). VidCrit (Chapter 6) lets video reviewers assign feedback to video content using spoken word and lets feedback recipients rewrite the labeled segments for ease of repeated search, browsing, and navigation tasks.

3.2 Video browsing and navigation

This thesis focuses on helping users complete high level searching, browsing and skimming tasks using structured text documents. Prior work supports video browsing and navigation based on keyframes, transcripts and related metadata.

Keyframe-based browsing and navigation

Several systems aim to facilitate video navigation and search by constructing visualizations of video frames, allowing users to click in that visualization to jump to the relevant point in a video [58, 21, 69] or enabling users to see more frames during scrubbing [91, 90]. Traditional video players included in web-based video players [52, 138, 145], editing tools [110, 53], and personal playback applications [114, 19, 111] also fit into this category as they enable users to preview frames while scrubbing along the timeline.

These techniques all rely on the viewer to recognize visual features of the desired location in videos. In contrast, our methods enable users to browse using text, which also surfaces information conveyed in the audio (e.g., speech, scene descriptions, sound effects).

Transcript-based browsing and navigation

A number of commercial video players (e.g., TED) feature a synchronized transcript. Berthouzoz et al. [27], Troung et al. [133], Leake et al [82] and Informedia [35] align a text transcript (or concatenated captions) to the video so that clicking on a word in the transcript navigates to the corresponding point in the video, while scrubbing the video highlights the corresponding part of the transcript. Our work builds on such transcript-based navigation tools.

In all projects, we use a transcript (or captions) aligned to the video to support the creation and alignment of structured text summaries. In Video Digests (Chapter 4) we use transcript-based navigation as a medium for creating higher level chapter and section summaries. In VidCrit, we use transcript-based navigation as a starting point for feedback-based navigation of a video (Chapter 6).

Metadata-based browsing and navigation

Many prior systems use domain-specific metadata to browse video – e.g., user interface events for software tutorials [89, 61, 37, 109], sports statistics for athletic events [102, 92], or computer vision and crowdsourcing approaches for informational lecture videos [66, 95, 66, 106, 72], how-to videos [74] and movies [116, 115, 94]. DIVA [86] and Chronoviz [54] align video with time-coded metadata (e.g. user annotations, notes, subtitles) to support exploratory data analysis of raw multimedia streams (e.g., for user studies).

We similarly develop domain-specific approaches for searching, browsing, and skimming. But, unlike prior work we focus on higher level searching, browsing, and skimming tasks by leveraging parallel text documents.

3.3 Video and audio editing

In this thesis, we do not edit video content but rather provide an index into the base video. However, work in video editing has considered methods for efficiently navigating the video content (e.g., in order to perform edits).

Traditional editing tools like Final Cut [53] and Premiere [53] typically use frame-by-frame navigation using video timelines. Several systems have used time-aligned transcripts to support audio/video editing through text manipulation operations [141, 35, 28, 118, 82, 133]. Such tools provide, for example, clip segmentation through editing markers in the transcript. Other metadata, such as annotations of actions and steps, can facilitate automatic shortening of how-to videos [38]. Or, logging [133] can provide a method for aligning b-roll to a narration track. Drawing from this body of work, we have focused the video digest editing interface and our VidCrit interface for revising comments on transcription-based interactions.

3.4 Video summarization

Automatically summarizing a video to include only the most salient content is a long-standing research problem. Truong and Venkatesh's [134] survey of work on this problem divides video summarization methods into two main approaches; *keyframe methods* identify a sequence of static frames that together represent the salient video content [135, 31, 22, 65], while *video skim methods* shorten the input video by removing non-essential content [124, 67, 130, 40]. Keyframe methods primarily focus on conveying the visual content of a video in static form and are not designed to expose any of the information content contained in the vocal audio track. Video skims concatenate important segments of a video into a shorter video, but still rely on opaque, timeline-based navigation of the video's content. In contrast, our work focuses on presenting the informational content of a video in a hierarchically organized chapter/section structure that supports browsing and skimming.

An alternative to algorithmic video summarization is to crowdsource the summarization task. Adrenaline [25] uses a crowdsourcing pipeline to extract representative keyframes from video segments. EpicPlay [127] uses viewers' interactions on social media to identify important moments in sports video. Lasecki et al. [79, 80] use non-expert crowds to transcribe videos and to describe activities in videos. Most related to our technique is the work by Kim et al. [73] which annotates steps in how-to videos. They use a Find-Verify-Expand technique to label steps in a how-to video and associates before and after images of each actionable step. Our work similarly relies on crowdsourced judgments to extract information from a lecture video.

Chapter 4

Video digests

4.1 Preamble

This chapter considers how to make speech-intensive video easier to search, browse, and skim. The structure of a video digest is chapters (i.e. a segment with the title of the chapter offset typographically) and section summaries (i.e. readable abstractive summaries of the transcript text) that create a navigable overview of the lecture content that is efficient to search, browse and skim. While the final video digest provides some flexibility in browsing (e.g., different levels of abstraction, or using the video player), the most flexibility in this method comes from the ability of experts to *author* video digests. Unlike our other navigational tools, authors can flexibly create different video digests for different scenarios. The authoring tool also lets authors leverage computational approaches (e.g., segmentation, crowdsourced summarization), and edit the output such that they achieve a shareable result. Finally, we show benefit of navigable structured text through an experiment comparing video digest to traditional video navigation methods (an aligned transcript and a traditional video player).

4.2 Introduction

Informative videos such as classroom lectures, seminar talks, and distance-learning presentations are increasingly published online. For instance, websites such as edX [1], KhanAcademy [8], and TED [15], offer thousands of informative video presentations on a wide variety of topics. Unlike live presentations, viewers can pause, replay, navigate, and alter playback speed to change the pace and structure of the video. The permanency of informative videos also provides a referencable resource for later review.

Bret Victor introduced a format for informational video presentations that is explicitly designed to helps viewer browse and skim a video presentation [136]. As shown in Figure 4.1, this format uses a textbook-inspired chapter/section structure to explicitly display the major themes in a presentation (the "chapters") as well as lower-level summaries of these themes (the "sections"). Specifically, each chapter corresponds to a topically-coherent segment of the video and consists of an embedded video player, a description (title) of the major theme in the segment, and a sequence of

Figure 4.1: A video digest affords browsing and skimming through a textbook-inspired chapter/section organization of the video content. The chapters are topically coherent segments of the video that contain major themes in the presentation. Each chapter is further subdivided into a set of sections, that each provide a brief text summary of the corresponding video segment as well as a representative keyframe image. Clicking within a section plays the video starting at the beginning of corresponding video segment.

section elements. Each section element provides a short text summary and representative keyframe for a video segment within the chapter-level segment. We call this format a *video digest*.

The visual design of such video digests exposes the content of a video at a topical level: viewers can browse the chapter titles to obtain an understanding of the major themes in the presentation and skim the short summaries and keyframes to gain a finer-grained understanding of the presented content. This format encourages dividing informative presentations into short, topically-coherent video segments which, as indicated by prior work, aids knowledge transfer and decreases dropouts for educational videos [93, 62, 73]. However, creating a video digest is a time-consuming process: authors must segment the videos at multiple granularities (chapter/section), compose section summaries, select representative keyframes, and create the final output display. Segmenting a video by topic often involves watching the video several times and scrubbing back-and-forth to find topic boundaries. Composing section summaries also typically requires re-watching a segment multiple times to make sure the main points are fully captured in the summary.

We present a set of tools to help authors create video digests by efficiently segmenting and summarizing the video through *transcript-based interactions.* The key insight of our approach is that much of the information in lecture videos is conveyed through speech. Therefore, our inter-

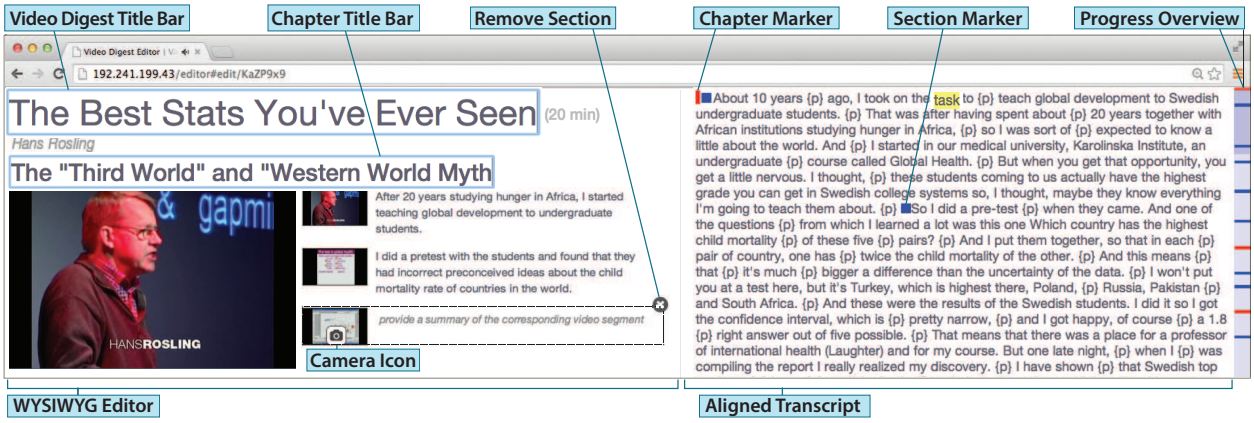

Figure 4.2: Our interface facilitates creating and editing video digests. The interface consists of two main panes: (1) An Aligned Transcript pane for navigating, segmenting and summarizing the talk and (2) a WYSIWYG Editor pane for adding chapter titles, summaries and keyframes for each section. Additionally, a Progress Overview scrollbar allows authors to view their segmentation progress and return to areas for refinement.

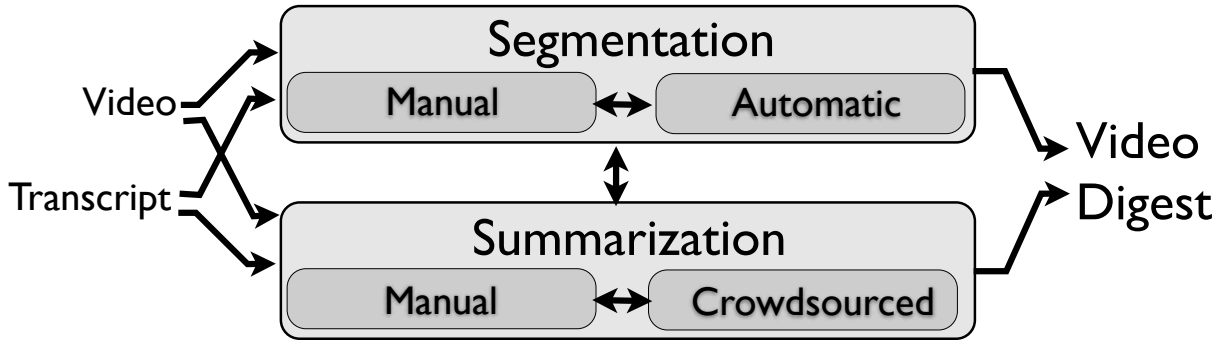

Figure 4.3: Given an input video and transcript, our authoring interface provides users with the ability to manually segment/summarize the content, automatically segment the content and crowdsource the summaries, or apply any combination of these two approaches.

face allows authors to navigate, segment and summarize the video using a time-aligned transcript of the speech. We also provide algorithmic tools for automatically segmenting the video and a crowdsourcing pipeline for summarizing the resulting segments. Authors can further refine these auto-generated digests in the authoring interface if necessary.

We use our tools to generate video digests for several kinds of informational videos, ranging from a TED talk to an online MOOC lecture. We compare manually authored digests to auto-generated digests and find that they both distill the informational content to the most important topics. However, the auto-generated digests are more verbose than the manually authored digests. Feedback from first-time authors suggests that our transcript-based authoring tools combined with

the automatic seeding greatly facilitate digest creation. We also conduct a crowdsourced study comparing video digests to timeline-based and transcript-based video player. We find that given 2 minutes to view lecture videos that are about 15 minutes long, viewers can recall up to twice as many of the key topics of the video using the digest format. At 8 minutes of viewing time, the differences between formats recedes. These results suggest that video digests support browsing and skimming of lecture videos better than the standard formats.

4.3 Creating a Video Digest

A video digest uses a textbook-inspired chapter/section structure to make the video easier to browse and skim (Figure 4.1). The chapter elements correspond to topically-coherent segments of the video that present a major theme, and the section elements segment the chapter into lower-level topic shifts. Each section provides a brief summary and representative keyframe for its corresponding video segment (Figure 4.1).

To produce such a digest, an author must complete the following tasks:

1. Segment the video into chapters
2. Title the chapters
3. Segment the video into short sections
4. Compose a text summary for each section
5. Select a keyframe for each section

Although most of these tasks can be interleaved, in practice we have found that video digest authors sometimes work bottom-up and create sections first (tasks 3-5), before grouping them into chapters (tasks 1-2), while at other times they work top-down creating chapter segments first (tasks 1-2), and then breaking them further into sections (tasks 3-5). Cycling between these two strategies is common [137].

Regardless of the strategy, each of the five tasks is time-consuming with current tools. Segmenting a video by topic often involves watching the video several times and scrubbing back-and-forth to find topic boundaries. Composing section summaries also typically requires re-watching a segment multiple times to make sure the main points are fully captured in the summary.

We have developed a set of tools to facilitate video digest creation (Figure 4.3). Our tools take a video and a corresponding transcript as input and lets users segment and summarize the video using a combination of manual, automatic, and crowdsourcing techniques. Users can interleave segmentation and summarization steps in any order. We first describe our video digest authoring interface and then present the algorithmic methods underlying this interface.

Video Digest Creation Interface

As shown in Figure 4.2, our video-digest authoring interface consists of two main panes: an *Aligned Transcript pane* (right) lets authors read the speech content, click on a word to navigate to the corresponding point in the video player, and mark chapter/section start points, while a *WYSIWYG Editor pane* (left) lets authors specify chapter titles, sections summaries and keyframes.

To support transcript-based navigation, segmentation and summarization, our interface relies on a time-aligned text transcript of the input video. When possible we obtain transcripts from the video source. For example, edX and TED provide transcripts of their talks online. Otherwise we use the crowdsourcing transcription service `rev.com` which accepts an audio file as input and returns a verbatim transcript for $1.25 per minute. We then time-align the transcript to the audio track of the video using the phoneme estimation and mapping technique of Rubin et al. [118, 146].

Chapter and Section Segmentation

To segment the video into topically-coherent chapters and sections, authors place chapter markers (red) and section markers (blue) in the Aligned Transcript pane using mouse clicks with modifier keys ('ctrl+alt' for chapter, 'ctrl' for section). Chapter markers denote the start of a major theme in the presentation while section markers denote less-significant topic changes within each chapter. Dragging these markers to different locations in the transcript changes the starting point of the chapter/section. In addition, clicking a section marker with the chapter modifier key creates a new chapter at that location, and the clicked section as well as all remaining sections in the original chapter are automatically moved into the new chapter. This operation splits the original chapter into two chapters at the clicked section marker. Conversely, clicking a chapter marker with the section modifier key removes the original chapter and appends all of its sections to the preceding chapter. This operation merges the clicked chapter with the preceding chapter. The Progress-Overview scrollbar on the right side of the Aligned Transcript pane represents the entire length of the video as a vertical bar. It shows the locations of all chapter/section markers and allows the author to quickly assess areas that need segmentation or refinement.

When the author places a new chapter marker in the transcript, our interface generates a new chapter in the WYSIWYG Editor with its video element cued to the location of the chapter marker. Similarly, when the author places a new section marker, our interface generates a new section keyframe and summary box in the WYSIWYG Editor pane at the appropriate location. By default, it fills the keyframe with the first frame of the corresponding video segment and places the cursor in an empty adjoining summary box. The WYSIWYG Editor automatically updates when the author drags a section/chapter start point to a different location or splits/merges chapters by clicking on chapter or section markers with the opposite modifier keys. Authors can delete sections by clicking the "Remove Section" button that appears when hovering over or modifying a section. Removing all sections from a chapter deletes the chapter.

In addition to these manual segmentation operations, the author can select a portion of the transcript and then invoke our automatic segmentation algorithm (see Algorithms Section) on the selected text, using a right-click menu.

Section Summaries and Keyframes

Authors compose section summaries directly in the summary boxes of the WYSIWYG Editor. Clicking on a summary box scrolls the transcript view to the corresponding text segment for quick reference. The author can replace the default keyframe by navigating the chapter's video player to the desired location and clicking a camera icon beneath the desired keyframe. Alternatively, an author can right-click on the summary box to invoke our crowdsourcing summarization pipeline (see Algorithmic Methods Section). The pipeline returns a crowd generated section summary and keyframe which the author can then refine if necessary. Finally, the author can set the video digest title and chapter-level titles by directly editing the text in the respective title bars.

4.4 Algorithms

Our system provides automated techniques for segmenting a video into topically-coherent units and obtaining summaries of such segments via a crowdsourcing pipeline.

Algorithmic Segmentation

Automatic text segmentation is a well-studied problem in natural language processing [68, 39, 49]. Eisenstein et al.'s [50] Bayesian topic segmentation (BSeg) algorithm is one of the leading techniques for segmenting speech-based text. BSeg is designed to group sequences of lexically cohesive text fragments into a segment – the text fragments can be any user-defined sequence of words in the text such as phrases, sentences or paragraphs.

The strength of BSeg is its ability to incorporate a variety of features such as cue phrases (e.g. "in conclusion", "therefore", "so", etc.) that might signal topic transitions. In pilot experiments[1] we found BSeg to outperform several other modern text segmentation algorithms [68, 87]. As a result, we use BSeg to automatically obtain both section- and chapter-level segments in our system.

BSeg produces a *linear* segmentation of the input text rather than the *hierarchical* chapter/section segmentation needed by our system. To obtain a hierarchical segmentation, we apply BSeg twice. First, we use the sentences from the original transcript as the input text fragments and BSeg returns a sequential grouping of these sentences into section-level elements. Next, we apply BSeg again, but we treat the output section-level elements from the first application as the input fragments to the second BSeg application. If summaries of the sections are available, we instead use these summaries as the input fragments to the second BSeg application. In either case this second application of BSeg groups the section-level segments into topically-coherent units that form the chapters of our digest.

Crowdsourced Section Summaries

We first exist prior work in crowdsourcing abstractive summaries, and then we discuss our method for crowdsourced section summaries.

[1] See supplementary material [106] for details on these experiments and our technique for setting BSeg's parameters.

Crowdsourcing abstractive summarization

Our work builds on several previous crowdsourcing techniques for abstractive summarization of text. Soylent [26] employs human language understanding to generate abstractive summaries to shorten text: different sets of crowd workers identify lengthy sentences, rewrite to shorten these sentences, and vote on the best edits. However, edits are limited to single sentences and the resulting summaries are not always coherent across different edits. Burrows et al. [33] develop a crowdsourcing pipeline for summarizing text documents that includes an automatic classifier designed to filter out poor summaries. Researchers have also presented techniques for using crowdsourced summaries of text documents to improve machine translation pipelines [34, 46]. We are inspired by these prior techniques and introduce a new crowdsourcing method to generate high-quality abstractive summaries based on text and video information.

Method

We have developed a crowdsourcing pipeline for obtaining section summaries. In our pipeline, one set of crowdworkers compose a summary and select a representative keyframe for each of the input sections. A second set of crowdworkers rank the summaries and keyframes based on quality. We then return the top-ranked summary for each section to the authoring interface.

In order to create browsable and skimmable video digests, each summary must concisely summarize the main topic of the corresponding section and use the same grammatical person and tense as the surrounding sections. We emphasize these goals in a set of guidelines we provide to the crowdworkers. We tell them that the summary should: (1) convey the main point(s) of the section, (2) omit non-essential details, (3) use the same tense (e.g. past, future) and grammatical person (e.g. first person, third person) as the section transcript, and (4) use concise wording, free of grammatical errors. In early experiments with the task, we found that workers often generated overly-detailed summaries. Based on these experiments, we added the guideline that (5) the summary should be less than three sentences in length.

For each summary task, we provide workers with the video and the aligned transcript cued to the section start point (Figure 4.4). We give workers access to the complete video and transcript so that they can build additional context when needed (e.g. to resolve ambiguous pronoun references in the section). We ask at least three crowdworkers to provide summaries and keyframes for each section. To ensure high-quality summaries, we then pipe these summaries into a ranking stage, where a different set of crowdworkers rank the quality of the summary-keyframe pair from the set of such pairs for each section. We ask these crowdworkers to base their rankings on the summary writing guidelines and to provide a short justification for their top-ranked selection. To reduce the cognitive load required to understand each section, we ask each crowdworker to summarize or ranks three consecutive sections.

We pay crowdworkers $0.60 to write three section summaries and select the corresponding keyframes. To incentivize high-quality work we offer $0.10 bonuses to the worker who writes the top-ranked summary for each section. Similarly we pay crowdworkers $0.60 to rank the summary-keyframe pairs for three sections.

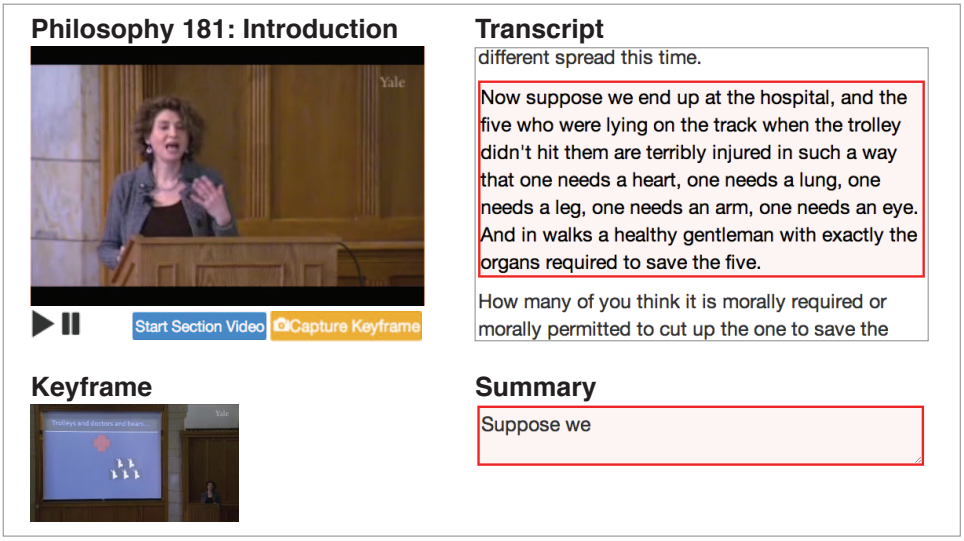

Figure 4.4: Crowdworker task for summarizing a section of the video. Workers can navigate the video using the timeline-based player (left) or the aligned transcript (right). The video is initially cued to the beginning of the section that must be summarized and the corresponding portion of the transcript is highlighted in red. Workers can select a keyframes by clicking on the *capture keyframe* button area, and they can write a text summary in the *summary* textbox.

4.5 Results

Figure 4.5 shows manual and auto-generated video digests produced using our tools. These results were generated using the following set of input videos (Table 4.1):

- *Philosophy 181: Introduction* by Tamar Gendler [57]: a 13.3 minute recording of an in-class philosophy lecture at Yale University (Figure 4.5A,B)
- *The Power of Prototyping* by Scott Klemmer [77]: a 13.8 minute introductory lecture from Coursera that uses a slide-based presentation (Figure 4.5C,D)
- *US History Overview: Jamestown to the Civil War* by Salman Khan [71]: a 18.5 minute overview of U.S. history from Khan Academy that uses a pen-based screencast presentation (Figure 4.5E,F)
- *The Best Statistics You've Ever Seen* by Hans Rosling [117]: a 19.9 minute high-production-quality, stage-based presentation (Figure 4.5G,H)

One of the paper authors created the manual video digests, and we created the auto-generated digests by combining our automatic segmentation algorithm on the entire transcript with our crowdsourced summarization pipeline. Table 4.1 shows the total time and cost required to create the manual (M) and auto-generated (A) digests respectively. We instrumented our authoring

	Gendler		Klemmer		Khan		Rosling	
	M	A	M	A	M	A	M	A
Time to create	40m	–	34m	–	41m	–	62m	–
Crowdwork cost	–	$19	–	$23	–	$35	–	$54
Num. of chapters	2	3	3	4	6	6	4	8
Num. of sections	20	26	14	13	17	16	20	40
Compression ratio	4.8	3.7	10.0	6.3	7.7	4.6	7.1	3.3

Table 4.1: Creating manual (M) and an auto-generated (A) video digests for four lecture videos required either authoring time (M) or payments to crowdworkers (A). We report the number of chapters, sections and compression ratios for each of the resulting digests.

tools to record the time spent performing each subtask involved in digest creation. In the manual case, we spent 48% of the total time reviewing the transcript and lecture video, 42% of the time composing section summaries, 6% of the time writing chapter titles, and the remaining 4% of time performing all other operations including placing segments. Overall, the time to create the digests manually was about 3-4 times the length of the input lecture. In the auto-generated case we recruited three unique crowdworkers to generate summaries and keyframes for each section and three more to rank each summary-keyframe pair. The total crowdsourcing cost was between about $0.50 and $1.00 per minute of the input lecture.

Table 4.1 also shows that the manual and auto-generated digests contain similar chapter and section counts for all of the lectures except Rosling. The Rosling lecture contains many short anecdotes that each use different vocabulary. In the auto-generated digest, BSeg segments the lecture based on the frequent vocabulary changes and produces twice as many chapters and sections as in the manual digest. Unlike BSeg, the manual digest author grouped together the anecdotes into higher-level concepts. Despite the differences in chapter/section counts, Figures 4.5G and 4.5H show that the two Rosling digests cover the same topics, but at different granularities. Other differences in the section-level segementations are shown in Figures 4.5I and 4.5J.

Finally we note that the manual digests usually contain less summary text overall than the auto-generated digests. Table 4.1 reports the compression ratio – the number of words in the original transcript divided by the number of words in the digest – for the lectures. Although the manual digests achieve higher compression ratios than the auto-generated digests, both condense the information compared to the transcript. The examples in Figures 4.5L and 4.5M suggest that crowdworkers tend to put more context into their section summaries which makes them longer than manually authored summaries. Because crowdworkers only summarize a small part of the lecture and cannot see the surrounding summaries, they may be compensating by repeating contextual information. In contrast, the author of a manual digest has access to all of the summaries and can eliminate such redundancies. Nevertheless, as demonstrated in Figure 4.5, the crowdsourced summaries do capture the the main concepts of each section similar to the manual summaries.

	User 1		User 2	
	M	E	M	E
Lecture	Klemmer	Gendler	Gendler	Klemmer
Time to create	54m	37m	30m	20m
Num. of sections	19	13	14	16
Num. of sections edited	–	6	–	4
Summary-edit keystrokes	3812	1355	1011	317

Table 4.2: In an informal evaluation two users created a manual video digest (M) and edited an auto-generated digest (E). We report the number of sections created for both (M) and (E), but only include the number of sections each user edited in the latter case.

4.6 Informal User Feedback

To gauge the utility of our video digest creation tools, we conducted an informal evaluation with two users (U1 and U2). We asked them to manually create a video digest and also to refine an auto-generated digest using our authoring tools. We examined the time they spent editing the digests and the number/type of edits they made. We also conducted a post-evaluation interview to gather qualitative feedback.

In the evaluation, U1 manually created a digest for Scott Klemmer's *Power of Prototyping* lecture and refined an auto-generated digest for Tamar Gendler's *Philosophy 181* lecture. U2 completed the opposite tasks for these two lectures. Both of these input videos are approximately the same length. Before starting the tasks, we presented each user with an example of a video digest, explained the chapter/section structure, and demonstrated the authoring interface. For the refinement task, we instructed the users to refine the digest so that it matched the quality of their manually created digest.

Table 4.2 shows that both users spent less time refining the auto-generated digest than creating the manual digest and edited fewer than half of the auto-generated section summaries. Both users performed fewer keystrokes when editing the auto-generated summaries, than when creating the summaries manually from scratch. When editing the auto-generated digests, modifying the sections was the dominant form of interaction. U1 focused on improving the flow between section summaries, while U2 mainly adjusted section boundaries. They rarely modified default keyframes or adjusted chapter boundaries when they were refining the auto-generated result. Although both users did edit the auto-generated summary text, they also provided positive feedback on the auto-generated summaries: U1 stated that the auto-generated summaries were "summarized in a way that I wouldn't think of myself, and I think what they did was correct and great." U2 noted that the auto-generated summaries were "on-target." U1 noted encountering a single incorrect summary, while U2 found one unnecessary segment boundary.

4.7 Study: Do Digests Support Browsing/Skimming?

We performed a comparative study to test the hypothesis that video digests afford browsing and skimming better than alternative formats. In our experiment we asked crowdworkers to watch one of four lectures (Gendler, Klemmer, Khan, Rosling) using one of the following formats:

- ***Manual:*** a manually created video digest using our tools.
- ***Auto:*** an auto-generated video digest.
- ***Video:*** a timeline-based video player.
- ***Script:*** a transcript-based video player.

We gave the crowdworkers a fixed length of viewing time (either 2, 5 or 8 minutes) and asked them to "quickly browse and skim" the content of the lecture. We then hid the lecture and asked them to provide "an approximately 5 sentence summary" of the main points covered in the lecture. We purposely did not give the crowdworkers enough time to watch the entire lecture so that they had to browse and skim its content to write a complete summary.

We asked 4 unique crowdworkers to summarize each combination of independent variables (lecture, format, viewing time) yielding 192 total summaries (4 lectures $\times$ 4 formats $\times$ 3 viewing time $\times$ 4 crowdworkers). We paid each crowdworkers \$0.90 for the summary plus a \$1.00 bonus if the worker obtained the highest summary score for the (lecture, format, viewing time) condition.

To evaluate the five sentence crowdworker summaries the first two authors of this paper worked together to manually build a *gold standard* list of key topics discussed in each of the four lectures. We then used these lists to score the number of topics covered in each crowdworker summary. Finally, we normalized the scores based on the total number of gold standard topics for each lecture.

Figure 4.6 shows the normalized scores for each format and viewing time aggregated across the four lectures. Using a Kruskal-Wallis test, we found significant difference in these scores when compared across viewing times (2 minutes: $\mu = 0.408$; 5 minutes: $\mu = 0.457$; 8 minutes: $\mu = 0.530$. $\chi^2(2) = 9.92, p = 0.007$). Further analyzing each viewing time, we found a significant difference in scores when comparing across the formats for 2 minutes ($\chi^2(3) = 18.31, p < 0.001$) and 5 minutes ($\chi^2(3) = 16.23$, $p = 0.001$) , but not for 8 minutes ($\chi^2(3) = 6.77$, $p = 0.080$). Pairwise Mann-Whitney tests with Holm-Bonferroni correction indicated significant differences in the following format pairs: At 2 minutes of viewing time, manual-video ($U(15) = 32$, $p = 0.002$), manual-script ($U(15) = 44.5$, $p = 0.009$), and auto-video ($U(15) = 58.5$, $p = 0.037$) are significantly different. At 5 minutes of viewing time, the manual-video ($U(15) = 56.5$, $p = 0.03$) and manual-script ($U(15) = 33.5$, $p = 0.002$) were significantly different.

Further analyzing the formats, we found a significant increase in scores when comparing across the viewing times for only the video format ($\chi^2(2) = 13.26$, $p = 0.001$). Pairwise Mann-Whitney tests with Holm-Bonferroni correction find significant differences in the following viewing time pairs: 2 minute-8 minute ($U(15) = 34.5$, $p = 0.001$) and 5 minute-8 minute ($U(15) = 67.5$, $p = 0.047$).

In short, at viewing times of 2 and 5 minutes, the video digest formats (manual and auto) allowed viewers to recall up to twice as many of the key topics than the video player formats (script and video). However, this effect diminished for the longest viewing time of 8 minutes. For the roughly 15 minute long lecture videos we tests, viewers could successfully recall many key topics after only 2 minutes of viewing time using the video digest; giving extra viewing time yielded little improvement that was not statistically significant. In contrast, with the standard timeline-based video player, summarization performance was low when viewers were given only 2 minutes, and gradually improved with additional time. Together these results suggest that both of the video digest formats – manually authored and auto-generated – facilitate browsing and skimming of informational lecture videos.

4.8 Limitations and future work

We see several promising directions for future work.

Ensuring consistency across crowdsourced summaries

Our current crowdsourcing pipeline does not ensure consistency between section summaries produced by different workers. Separate workers may include redundant information or use pronouns that are ambiguous given previous summaries. One solution might be to include an additional stage in the crowdsourcing pipeline where new crowdworkers check multiple consecutive summaries for overall consistency. These crowdworkers could also provide titles for the video digest chapters as our current system does not produce such titles automatically.

Support for highly-technical content

We tested our automatic pipeline on four lectures that are accessible to a broad, well-educated audience. However, we have not tested our tools with lectures that require specialized knowledge, or highly-technical content where crowdworkers may not have the necessary background to write summaries. One fruitful direction for MOOC style lectures may be to ask students who choose to watch a lecture to write the summaries, e.g. students in a graduate-level quantum mechanics course. Creating summaries may help the students learn the material and also generate high-quality summaries for technical material.

Use video data in algorithmic segmentation

Our segmentation algorithm only uses the transcript to segment the video. Future work could incorporate visual and audio information to improve video segmentation. It may also be possible to use viewer interaction data to automatically infer segmentation points in the video in the manner of Kim et. al [75].

4.9 Conclusion

We have presented a set of tools for creating video digests; a new format for informational talks that exposes the structure of the content via section-level summaries and chapter-based grouping. We provide a transcript-based authoring interface and explore techniques for automatically segmenting and summarizing an input video. An informal evaluation suggests that our tools make it much easier for authors to create video digests, and a crowdsourced experiment indicates that the video digest format affords browsing and skimming better than alternative video presentation interfaces. Next, we'll explore methods for searching, browsing and skimming videos that make use of domain-specific knowledge to further automate the process of generating a structured text representation.

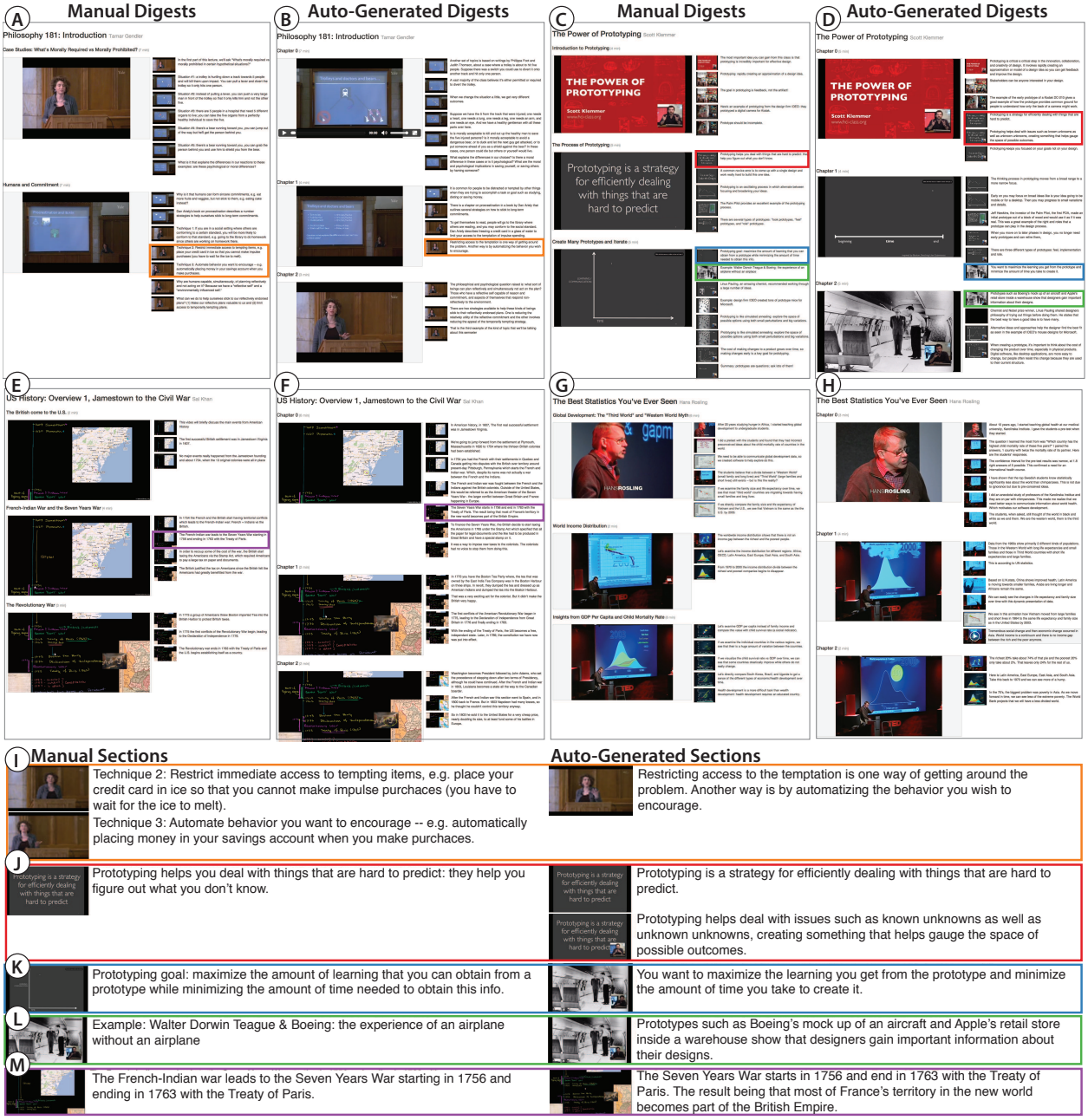

Figure 4.5: Manual and auto-generated video digests for four lecture videos Gendler (A,B), Klemmer (C,D), Khan (E,F) and Rosling (G,H). Differences between the manual and auto-generated results are highlighted below (I-M). Example (I) and (J) show differences in segmentation where two sections in the manual digest are combined into one section in the auto-generated digest and vice-versa. Example (K) shows how sections summaries can be very similar between the manual and auto-generated digests. However, examples (L) and (M) show that the manual digests often include more succinct summaries than the corredponding auto-generated digests.

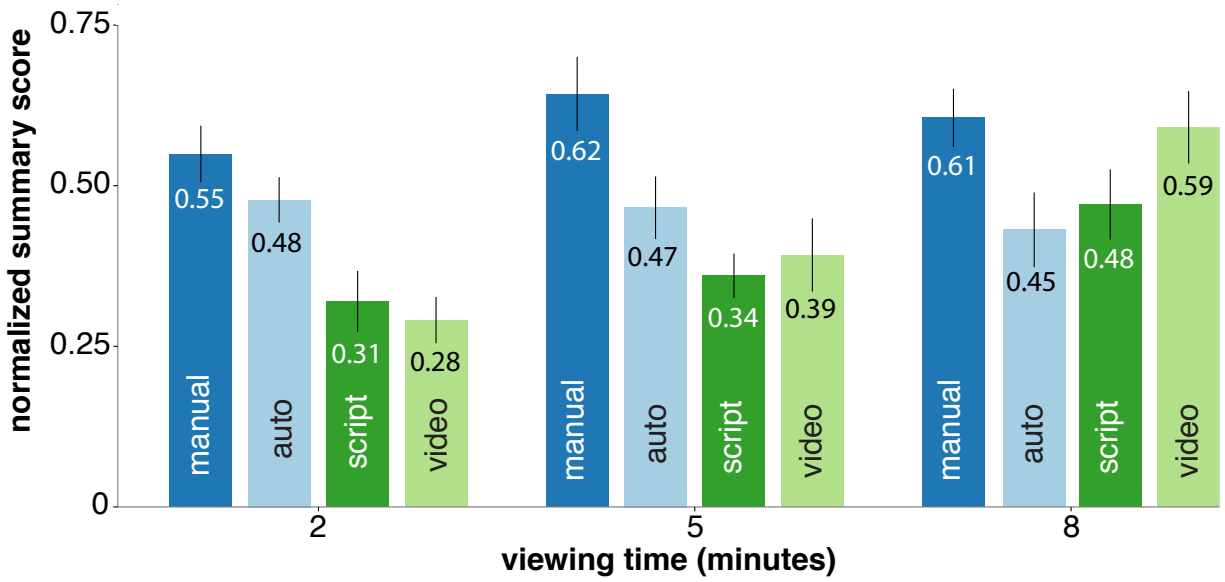

Figure 4.6: Crowdworkers viewed a video lecture in one of four formats (manual, auto, script, video) for 2, 5 or 8 minutes and then wrote a summary of the presentation. We scored these summaries using a gold standard topic list. Each bar shows the mean and standard error of the scores for each condition.

Chapter 5

Sceneskim

5.1 Preamble

SceneSkim makes use of domain-specificity to fully automate the creation of a flexible and efficient structured text navigation tool for film. While video digests provided flexibility in authoring, SceneSkim transfers flexibility to the end user by providing three parallel representations aligned to one another and the original video. Using documents familiar to practitioners, we can leverage their domain knowledge to understand what questions can be asked of the system. We investigate the flexibility of our system through answering a set of existing film studies questions, and letting practitioners answer novel questions using the tool.

5.2 Introduction

Searching for clips in film is a crucial task for film studies researchers, film professionals and media editors who analyze, share and remix video clips. For example, film studies researchers may search for clips containing a particular action, prop, character, or line of dialogue, in order to discover patterns in films. Film professionals find existing examples of settings, props, character performances, and action sequences in order to inspire new projects and communicate desired visual attributes (e.g. animation style, lighting, sets). Movie fans remix existing Hollywood movies and TV shows into "supercuts": montages of repeated elements from existing films such as words, phrases, or cliches. However, like informational videos videos, searching and browsing within movies is a time-consuming task. Consider a user who wishes to analyze the context and appearance of lightsabers, in the *Star Wars* movie series. If she knows the films already, she might try to navigate to remembered scenes using DVD chapter menus, or by scrubbing through video timelines. Otherwise, she might have to simply watch all of the movies, taking notes whenever a lightsaber appears. The 4 practitioners we interviewed primarily search for clips using those same approaches. Some video viewing interfaces allow users to search and browse videos by text transcripts [15, 27, 54, 86] or caption files [35, 81], these documents do not always contain contextual information (e.g. locations, props, actions, or names of characters speaking the dialogue) that may

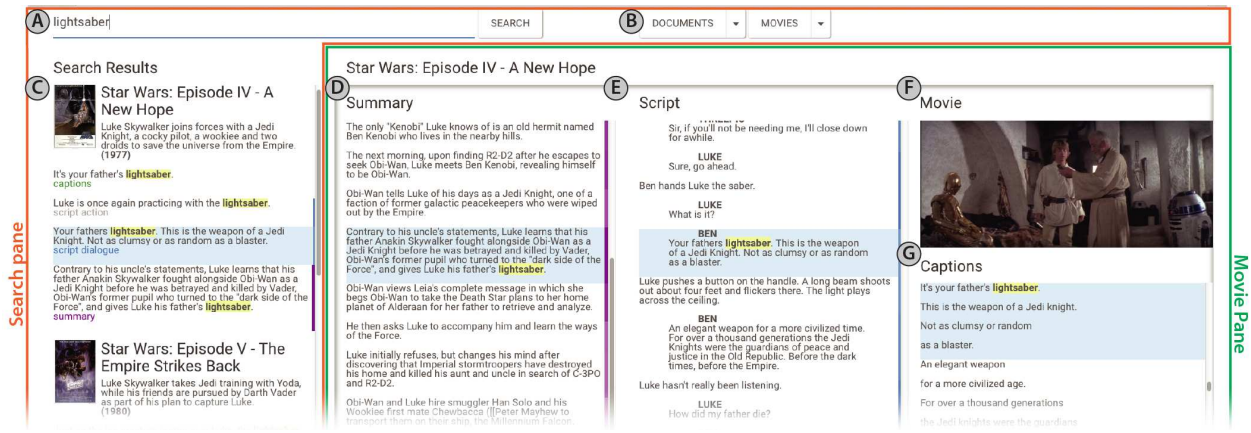

Figure 5.1: The SceneSkim interface consists of a *search pane* for finding clips matching a query and a *movie pane* for browsing within movies using synchronized documents. The search pane features a keyword search bar (A), search filters (B) and a search results view (C). The movie pane includes the synchronized summary (D), script (E), captions (G), and movie (F).

be relevant to these searches. In addition, it can be difficult to browse for a specific scene using transcripts because they lack visual context.

For movies, there often exist documents containing structured text with relevant features for search (Figure 5.2): Movie scripts identify speakers and contain a written description of the dialogue, actions and settings for each scene (Figure 5.2C). Online movie databases include user generated plot summaries that describe important moments, high level events (e.g. escapes), and character traits (e.g. a hostile character) (Figure 5.2D). We call these related datasets *parallel documents*. However, although they contain a wealth of contextual information about the same underlying film, the data in parallel documents are generally not explicitly linked to each other or to the underlying film.

We developed SceneSkim to help users search and browse movies more efficiently by jointly exploiting all of these data sources. SceneSkim automatically aligns the captions, script and plot summary to the audio-visual content of a film. SceneSkim also aligns each parallel document to one another, enabling synchronized browsing of the captions, script and summary for easy access to details or context at any point in the film. SceneSkim leverages the structure of the related documents, allowing users to search over specific document features, such as lines by a particular character, or scene locations. Our implementation introduces new techniques for aligning the summary to the script and the words in the captions to the audiovisual content. In particular, we use unique terms to inform an ordered alignment between the summary and script, and we use caption timestamps to aid forced alignment between caption words and movie audio.

Returning to our lightsaber example, Figure 5.1 shows how SceneSkim lets our user easily discover the complete context of lightsabers in the *Star Wars* films. First, the user types "lightsaber" into the search bar (A). The search results (C) reveal caption, script and summary sentences describing the appearances of lightsabers in chronological order within each film. The first entries reveal the main character's earliest encounter with a lightsaber, which they views in the video panel

(F) by clicking on one such sentence. From there, the user can either browse through that scene by scrolling and clicking in the script (E), or navigate to dialogue about lightsabers with the captions (G), or continue to scroll in the search results (C) to other appearances of lightsabers.

We used the SceneSkim interface to answer queries (e.g. for character appearances, locations, dialogue, and actions) described in existing film studies research [3, 32, 45, 121]. For instance, we were able to manually estimate total screen time of lightsabers across three *Star Wars* movies in about 20 minutes. In an informal evaluation with three film studies researchers, the researchers were able to formulate and answer novel questions using our system. The users reported they would use such a tool for day-to-day tasks in research and teaching, and as an exploration tool. Users positively commented on our system's usability and suggested new features.

5.3 Definitions

Each parallel document provides a unique type of information, enriching searching and browsing capabilities. In this section we define background terminology related to these documents.

A movie, once shot and edited, is comprised of a sequence of continuous videos (i.e. *shots*) that span several locations (i.e. *settings*), and a synchronized audio track (Figure 5.2A). *Captions* transcribe the dialogue, sound effects, and relevant musical cues in the movie (Figure 5.2B). A *script*, or screenplay, is a written description of the film that includes the dialogue, actions, and settings for each scene (Figure 5.2C). At the beginning of each *scene*, screenwriters typically provide a *slug line* which specifies the setting. *Action* lines describe character actions, camera movement, appearance, and other details. The *dialogue* proposes what each character will say. However, the planned dialogue in the script and the transcribed dialogue in the captions do not match exactly: Editors may remove lines or scenes from the final film, and actors may improvise dialogue. The *character name* specifies which character will speak the draft dialogue. *Plot summaries* give a high level overview of main events, character traits and locations in the movie (Figure 5.2D). Such plot summaries are typically written by viewers or critics. Plot summaries contain higher-level event descriptions than contained in the script – which describes scene by scene actions – or captions – which directly transcribe the movie dialogue. Unlike the captions and script, summaries leave out events that are not central to main plot lines.

Current practice

To learn about current practices for searching and browsing in movies, we interviewed two individual film studies scholars and a film studies research group, as well as two visual effects professionals. We also analyzed a corpus of 101 supercuts [20].

Film studies researchers typically search to analyze sets of clips and find patterns. In particular, they search for specific actions, props, locations and characters in order to study audio and visual attributes of the corresponding clips. These researchers often return to clips they have seen before for further review. They identify text results of interest through Web search and then both watch and scrub through films to locate clips of interest. One researcher noted that it is easy to

miss short events while scrubbing. The research group mentioned they would like to search dialogue by keyword (e.g. to analyze scenes where particular slang terms occurred), or by performing characters (e.g. to study prominence and time given to a character).

Because film studies researchers frequently study visual attributes of scenes, text-based search over scripts alone is insufficient, but accessing film scenes through text search could be very helpful. For instance, one researcher wanted to study the body language of actors performing a common action. Another researcher wanted to study if communication technologies (e.g., cell phones) appeared as focal or peripheral in different films.

Film professionals search for locations and props in order to design new sets or create concept art. Such professionals also return to particular parts of films they have seen before. In addition, they search for character movements and dialogue to cast actors, create new characters, and inform the design of animations. Our interviewees reported that they search for clips on the Web, hoping that someone had uploaded a scene to YouTube or other video sharing sites. As a fallback, they search through their personal collections using DVD chapter menus. After locating clips of interest, they save the clip to review or send the clip to team members.

We also categorized **supercuts** published on supercut.org to infer the practices of media editors and remixers. We found that out of the first 101 supercuts (in alphabetical order by movie title), 60 were based on occurrences of some word, phrase or vocal noise. 28 were based on occurrences of some action, 5 were based on dialogue by a particular speaking character, and 4 were based on appearances of a given character.

To summarize, film studies researchers, visual effects professionals, and media editors all frequently search within films for clips matching a particular query, or browse to return to a particular scene in a movie. Specifically, users search for clips matching the following types of queries:

- Performances by a specific character or sets of characters (e.g. to closely study performance by character, or to watch main events containing the character)
- Locations (e.g. city skyline, living room)
- Actions (e.g. playing video games, car chase)
- Objects (e.g. cell phones, laptops)
- Words or phrases in dialogue (e.g. slang terms)

5.4 Prior movie navigation interfaces

MovieBrowser [94], VideoGrep [81], and the system by Ronfard *et al.* [116, 115] address the problem of browsing for specific clips in movies. MovieBrowser uses computer vision and audio

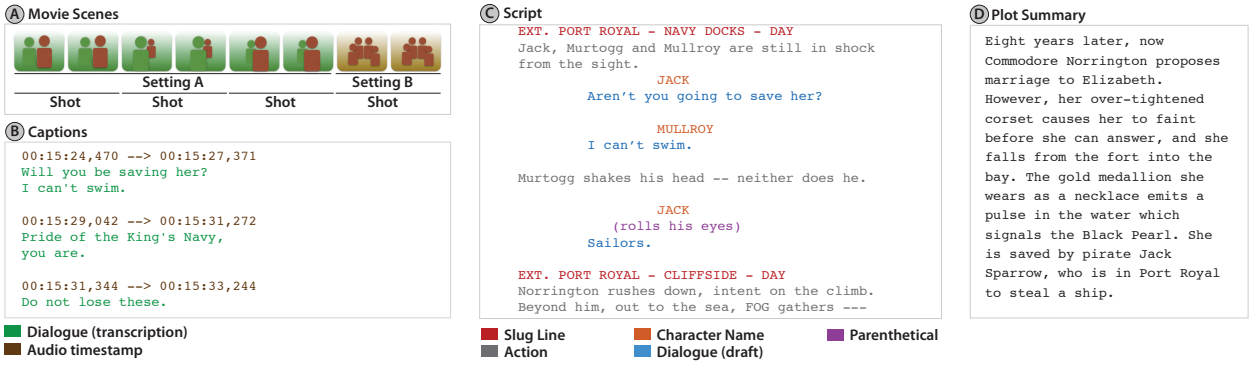

Figure 5.2: Different documents describe different aspects of a movie: The script (C) contains locations, character names, actions, parentheticals and draft dialogue of the movie (A), in which scenes comprise multiple shots. The captions (B) contain timestamped dialogue as spoken, while the summary (D) contains a high-level description of the plot.

processing to predict action sequences, montages, and dialogue and displays clip classifications on the movie timeline. Although this work allows users to browse clips by clip type, it does not allow browsing by other aspects of clip content (e.g. dialogue, plot). VideoGrep is a command line tool allows users to search videos using timestamped captions in order to create supercuts. But captions contain about 10 words on average, whereas our system facilitates individual word-level indexing, and searching over plot and visual descriptions.

The system by Ronfard *et al.* [116, 115] is most similar to our work. It offers browsing and search capabilities for *The Wizard of Oz* using an exact shot-by-shot script synchronized to corresponding shots in the DVD. Shot-by-shot scripts describe each shot and transition in detail, but such shot-by-shot scripts are rare (we were only able to find one other shot-by-shot script besides *The Wizard of Oz*). In addition, most films deviate from their source screenplays. We create a system that processes approximate scripts in a common format, and surfaces possible misalignments using confidence markers. Our system also allows users to search across multiple movies simultaneously, and uses the captions and summary along with the script to aid searching and browsing tasks.

5.5 SceneSkim interface

Motivated by these tasks, we developed the SceneSkim interface to support searching for clips that match a query and browsing for specific clips within a movie. We acquired complete caption, script and plot summary sets for 816 movies from existing corpora [131, 99] and by scraping the web (for more detail on this set, see Appendix A). We also purchased and loaded seven movies (*Star Wars* episodes 4-6, *Chinatown*, *Taxi Driver*, *The Descendants*, and *Pirates of the Carribean: Curse of the Black Pearl*) to demonstrate interactions. The interface supports searching and browsing this data set through two main components (Figure 5.1): the *search pane* and the *movie pane*.

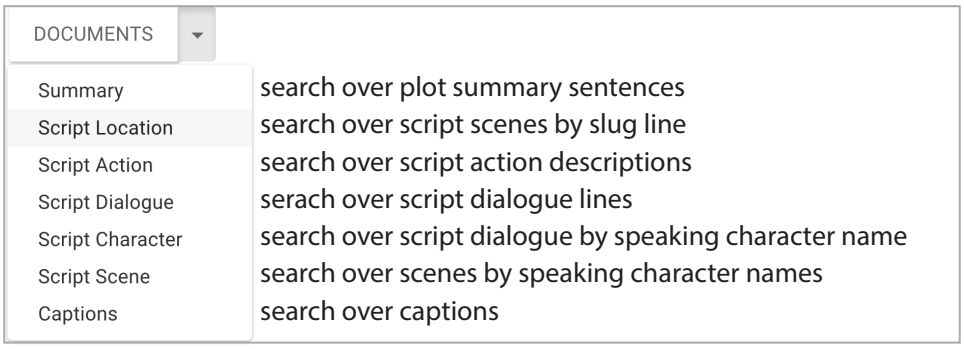

Figure 5.3: Users can select one or more entity types to search over using the "Documents" dropdown menu, from Figure 5.1b.

Search pane

The search pane enables faceted search of movie clips through the keyword search bar, search filters, and search result view.

Keyword search: The keyword search bar allows users to specify keywords and concatenate them with the boolean expressions AND and OR. (Figure 5.1A).

Search filters: The search filters in the "Documents" dropdown menu (Figure 5.1B) allow users to restrict keyword searches to specific entity types. The user may select one or more entity types to search over (Figure 5.3).

Search results: Within the search result pane (Figure 5.1C), we first sort results by movie (Figure 5.4). Each movie heading contains a movie poster, title, short plot summary, and release year. We then sort results by entity type (e.g. caption, script dialogue, script location etc.) and then in chronological order.

Each result shows a text snippet with the search terms highlighted, the result type label, and a confidence bar. The color of the confidence bar shows the likelihood that clicking on a script or summary result will jump to the correct part of the film. Darker colors represent better alignments. When the user clicks a search result, the movie pane scrolls all documents to the corresponding sections and plays the movie from that point onwards.

Movie pane

The movie pane allows users to browse within a movie using synchronized documents (Figure 5.1). From left to right, the movie pane displays the summary, script, and movie with captions below. We display the summary broken into summary sentences, the script broken into lines, and the captions broken into timestamped phrases. We align each document to all others, and by clicking on any document the user can view the corresponding sections in other documents. For example, clicking

Figure 5.4: Search results for searching "falcon" across all entity types. **Caption** results show caption phrases, indexing into the movie at the word level, **script dialogue** results contain full dialogue lines, **script location** results show scene headings, and **summary** results show high level events. The last result is a **script action** result. As the Falcon is not a character name, we do not see **script character** results, or **script scene** results. The search results are also shown in (Figure 5.1C)

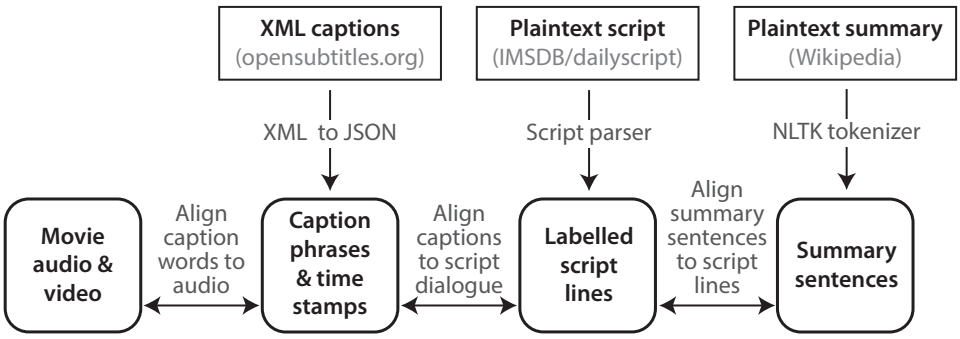

Figure 5.5: To faciliate searching and browsing using synchronized documents, SceneSkim loads the movie, captions, scripts, and summaries from outside data sources then finds alignments between pairs of documents and the caption and the movie.

on a summary sentence will cause the captions and script to scroll and highlight the corresponding script lines and caption phrases. The movie will also play starting at the estimated start time for the summary sentence. Similarly, the documents scroll in synchrony and the movie plays when a user clicks on a script line or caption word.

As in the search results, each script line and summary sentence also includes a color bar at the right margin of the column, which is used to visualize the confidence of the match between document regions and the movie. For example, sections of the script omitted from the final movie will display a white bar indicating no match. Sections with high confidence matches will appear dark blue, and lighter blue bars indicate low match confidence.

5.6 Algorithms

Our interface relies on several algorithms to let users search and browse movie clips using parallel documents (Figure 5.5). To allow users to access movie clips via caption words, we find a word-level alignment between the caption dialogue and the film. To support browsing the movie based on the script, we parse the script to identify the dialogue and then align the script dialogue to the caption dialogue. We facilitate browsing the movie and script through high level events by aligning the script to the plot summary.

Prior work in aligning movie metadata

SceneSkim facilitates searching and browsing over aligned captions, scripts, and summaries. Informedia [35] finds a word-level alignment between captions and broadcast news video by concatenating the captions and aligning the resulting transcript captions to a speech-to-text transcript of the film (similar to Berthouzoz *et al.* [27]). However, speech-to-text methods expect clean speech audio as input, and thus may fail when the audio track includes non-speech sounds characteristic of movies, such as background music and sound effects. Instead we use the alignment method of Ru-

bin *et al.* [118] to align caption phonemes to audio features directly, leveraging caption timestamps to remove extraneous non-speech noise.

Prior work also aligns scripts to movies and TV shows in order to identify characters [51, 112] and actions [30, 78, 44] in the video. We build on this work, using the edit distance technique first described by Everingham *et al.* [51] for aligning script dialogue to caption dialogue. Tapaswi *et al.* [129] and Zhu *et al.* [147] align movies to their original books, but we focus on aligning movie scripts to plot summaries. Tapaswi *et al.* [128] align plot synopses to the caption dialogue and identify character faces in TV episodes. We align plot synopses to scripts which contain information about locations, objects, and actions not mentioned in the caption dialogue. The prior work in this area develops algorithms primarily for machine learning purposes, and does not build user interfaces for search and browsing that take advantage of such alignments.

Caption to film word-level alignment

Captions consist of a sequence of transcribed phrases of dialogue, but are not always segmented or labelled by speakers. For instance, in Figure 5.2B the timestamp of the first caption spans lines for two characters: Jack asks "Will you be saving her?" and Mullroy answers "I can't swim." (Character names are shown in corresponding script lines in Figure 5.2C.) Thus, to enable users to search and view dialog lines uttered by individual characters, we require a finer-grained synchronization than timestamped phrases. Our system therefore computes a word-level alignment between the captions and the movie.

Our system employs Rubin's implementation [119] of the Penn Phonetics Forced Alignment (P2FA) algorithm [146] to find a word-level alignment between speech audio and the caption text. P2FA first generates expected phonemes from the caption words, and perceptual linear prediction features from the audio. Then, P2FA computes an alignment between the phonemes and audio features using a Hidden Markov Model. However, this method alone does not work well for movies, because movies contain many non-speech sounds—like sound effects and music—that falsely align to phonemes. To overcome this limitation, we constrain alignment to audio containing speech using the caption timestamps. Although timestamps in caption files can be inaccurate, we use them only as approximate markers: Our system groups caption phrases with timestamps less than .25s apart (the approximate length of a spoken word), and then applies P2FA to align the words in the grouped caption text to the corresponding audio.

Caption-to-film alignment accuracy: We randomly sampled 8 one minute sections from one of the movies in our dataset and created ground truth alignment for the 702 words included in the sampled sections. The start of each word deviates from ground truth by an average of 0.067s ($\sigma = 0.28$) compared to 1.1s ($\sigma = 0.77$) using caption time stamps alone.

Parsing Scripts

Authors format screenplays according to industry standards (Table 1) as detailed in the AMPAS screenplay formatting guide [2]. We exploit this standard to assign action, character, slugline, dialogue, and parenthetical labels to each script line (Figure 5.2A). However, the formats of the

Item	Properties
Slugline	All caps
	Interior/exterior designation
	Left margin: 1.5 in.
Action	Left margin: 1.5 in.
Character	All caps
	Left margin: 4-4.2 in.
Dialogue	After character or parenthetical
	Left margin: 2.5 in.
Parenthetical	Enclosed in parenthesis
	After character or dialogue
	Left margin: 3 in.

Table 5.1: This table describes the standard format for different types of script elements. The format distinguishes elements using spacing, capitalization, and specific labels such as the interior (INT) or exterior (EXT) labels used in slug lines.

scripts in our dataset deviate from industry standards, because some scripts have been converted from the original PDFs into text documents by third parties, losing some of the original formatting in the process. Our parser relaxes standard requirements by using relative spacing between script elements to determine labels instead of exact margin size. In particular, we assume that the margins of character names are not equal to slugline margins, so that the parser can distinguish these attributes. We additionally assume that the dialogue has a different left margin than the action descriptions. We discard scripts that do not meet these relative spacing requirements.

The output of our parser is a label (e.g. character name, slugline, dialogue, parenthetical, or action) for each script line. The supplementary material contains a detailed description of our script parser.

Parser accuracy: We evaluated our parser by creating ground truth for a randomly selected set of 20 non-empty lines in the original text document from each of 20 movies. We find our parser assigns labels to the text lines with 95.7% accuracy. Most errors are due to our parser's reliance on spacing to classify line type: In this evaluation, the parser mistakenly classified some dialogue lines as action lines in a script containing widely varying spacing for the dialogue lines and character names.

Script dialogue to caption dialogue alignment

Our system matches script dialogue lines to their corresponding caption dialogue phrases in order to allow users to browse the captions and the movie using the script. These two documents do not match exactly—captions contain improvised lines not present in the script; and scenes described in scripts may have been cut from the final movie.

Following the method of Everingham *et al.* [51], our system finds an alignment between the script dialogue and captions documents using the Needleman-Wunsch (NW) algorithm [97]. NW allows for insertions and deletions, which account for most differences between scripts and captions. Our system uses NW to find the highest scoring alignment between the sequence of all script dialogue words and the sequence of all caption dialogue words, where a matching pair counts for 1 point, and any mismatching pair, insertion, or deletion counts for -1 point.

Our interface then uses this alignment to play back dialogue at a line level. When users click on a script dialogue line, the system plays caption lines where one or more words aligned with words in the script line. For instance if a user clicks "Aren't you going to save her?" in the script (Figure 5.2C), the system would play the caption line "Will you be saving her"(Figure 5.2B) because the words "you" and "her" match.

Dialogue-to-caption alignment accuracy: To evaluate how well the script dialogue and caption dialogue align in our dataset, we counted the number of caption phrases which had at least one matching word with a script line for 13 randomly selected films, and found an average of 81% ($\sigma = 9$) of caption lines had a matching script line.

To help users navigate around parts of the script that do not match to the final film, our interface includes a confidence bar displaying measures of correspondence. We set the correspondence score by calculating α for each script line and caption phrase match. Then, we set the opacity of the dark blue confidence bar between 0 (for $\alpha = 0$) and 1 (for $\alpha >= 0.5$).

Summary sentence to script alignment

To align the summary to the script our system first splits the summary into sentences using the NLTK sentence tokenizer [29]. Then, the system produces overlapping windows of script lines of length K, which we set to 14 based on alignment accuracy with a small training set. Next, the system finds an ordered alignment between script windows and summary sentences maximizing the TF-IDF [125] similarity of matched pairs.

TF-IDF is a similarity measure used in data mining to compare two documents in the same corpus. In that context, each document is represented as a vector in which each element is the frequency of a term in that document, divided by the frequency of the same term in the corpus as a whole. The similarity between two documents is then computed as the cosine distance between the corresponding vectors. In our context, the TF-IDF for a script window is a vector in which each element is the frequency of a term in that script window, divided by the frequency of the same term in the entire summary and script. The TF-IDF vector for a summary sentence is a vector of elements in which, for each term, the term frequency in the sentence is divided by the term frequency in the entire summary and script.

Thus, for all N script windows $(w_1, w_2, .., w_N)$, and M summary sentences $(s_1, s_2, .., s_M)$ our system finds the TF-IDF vector for each script window and summary sentence. The system constructs matrix T where $T_{i,j}$ is the cosine similarity between the TF-IDF vector for s_i and the TF-IDF vector for w_j. To find an ordered alignment between summary sentences and script windows, our system again uses use Needleman Wunsch, setting the cost, $C_{i,j}$, of matching any pair

to $T_{i,j} - mean(T)$ and the cost of skipping a summary sentence or script window to -0.1. We chose -0.1 to balance between allowing insertions/deletions and allowing matches with low TF-IDF scores. In particular, we prefer an insertion or deletion over a match if the match score is less than the mean TF-IDF score minus 0.1. Our TF-IDF-based summary alignment algorithm works better for movies that contain many distinct objects, actions, characters, and location terms for different parts of the movie (i.e. "action-driven" movies), than for movies that focus on character development (i.e. "character-driven" movies).

Summary-to-script alignment accuracy: To evaluate the summary sentence algorithm we created ground truth summary-to-script alignments for four movies randomly selected for the most popular genres in our dataset (e.g. drama, action, comedy, romance). For the movies *King Kong*, *Code of Silence*, *American Graffiti* and *Punch Drunk Love* we find that the algorithm's predicted alignment matches at least one ground truth line 82%, 83%, 75% and 57% of the time respectively. In practice, this summary alignment aids users in locating specific clips in the movie by navigating to a nearby region within a long movie. After using the summary alignment to get close to the location of interest, users can then use finer-grained script or caption navigation to find the desired clip.

5.7 Dataset

We obtained complete caption, script and plot summary sets for 816 movies by joining three document databases: the caption corpus collected from opensubtitle.org by Tiedemann *et al.* [131], the Wikipedia plot summary corpus collected by by Bamman *et al.* [99], and a script corpus we generated by scraping dailyscript.com and imsdb.com for all scripts in plain text format.

Specifically, we join the script corpus and summary corpus using provided movie titles, then find the IMDB identification number for each movie in our dataset using the OMDB API [55]. For each movie, we use the movie's IMDB identification number to find the corresponding caption file in Tiedemann's caption corpus [131]. We remove all movies that lack a caption file, script file, or summary file, leaving us with 1002 unique movies with all three documents in our dataset. We then remove all movies that do not have a script in the format accepted by our parser, which yields the 816 unique movies in our dataset.

This dataset is biased towards recent movies in wide public release, as captions and summaries are more readily available for such movies. Although our dataset contains movies released between 1928 and 2013, half of the movies in the dataset were released in 1999 or later, and 90% of the movies in our dataset were released in 1979 or later.

Because movies themselves are not freely available, we purchased seven movies (Starwars 4-6, Chinatown, Taxi Driver, The Descendants, Pirates of the Carribean: Curse of the Black Pearl) to demonstrate interactions.

5.8 Evaluation: Informal user evaluation

We conducted an informal evaluation with three film studies researchers in order to get feedback about how SceneSkim might support their work. We recruited the film studies researchers by e-mailing all film studies graduate students at Berkeley. We first posed four concrete search tasks[1]. In a subsequent exploratory section, users formulated their own questions and answered them using our system. The concluding interview gathered qualitative feedback about the system. Our search tasks focused on questions about characters, costumes, relationships, and settings in the original *Star Wars* trilogy. For example, we asked: *Given that Chewbacca doesn't speak English, how does the film use growls to convey relationships between Chewbacca and other characters?*

Users successfully answered all the questions in the tasks section using our system in 1-3 searches per question. In the exploratory section, U1, U2, and U3 used our system to answer new questions[2]. We invited users to use all movies in our system, including the majority of the corpus that included only parallel documents, without the source videos. All three users chose movies with source video. U1 asked in which circumstances the sentence "Are you talking to me?" occurs in *Taxi Driver*. She searched for the words "you talking to me" over **script dialogue** and **captions** only. She found that only the main character uses this phrase, and that it only occurs in the captions and never in the script, suggesting that the actor either improvised the phrase, or it was added in a later script draft. Because the line did not occur in the script, U1 watched the clip for each caption line to definitively identify the speaker each time. U2 wanted to know if Jack Sparrow's eccentric body movements in *Pirates of the Caribbean* were explicitly written into the screenplay, and if so, how they were described. She searched for "Jack" in **scenes** to retrieve all scenes in which the character occurs, watched the scenes to find where Jack makes these movements, then read the script corresponding to those scenes. She did not see a consistent description of these movements in the script, which suggests that they are most likely the actor's invention. In the interview, all three users expressed strong interest in using the tool in the future, inquired about plans for releasing the tool, and commented positively on its usability.

U3 wanted to know how Tarkin dies in the first *Star Wars* trilogy. He searched for "Tarkin" in **script characters** to find all of Tarkin's lines and navigated to Tarkin's last speaking line. Then, he navigated beyond this using the script to the next time Tarkin appeared in an action description. He clicked on the action description to view Tarkin in a space station shortly before the space station explodes.

In the interview, all three users expressed strong interest in using the tool in the future, inquired about plans for releasing the tool, and commented positively on its usability. U1 explained *"[the system] is just really useful. [...] I feel like I would use this a lot, for research, for teaching and for looking at scenes."* U3 mentioned *"if I had the tool open in front of a class room, it would be an incredible teaching tool. [...] You would be able to respond to student's questions, suggestions and requests on the fly."* All three users inquired about plans for releasing the tool and commented positively on its usability.

[1] Our search tasks focused on questions about characters, costumes, relationships, and settings in the original *Star Wars* trilogy

[2] We explain U2's query here, U1 and U3 can be found in the paper [103]

Users found that each of the different search and browsing features successfully supported particular tasks. Summaries were useful for navigating familiar movies, e.g. for revisiting clips repeatedly. Our users also mentioned uses we had not considered: U1 explained that having captions and script aligned to the movie facilitates pulling quotes for research essays from both documents. U2 explained that having all of the parallel documents aligned to each other encouraged new ideas: *"I like that the four types of information are available simultaneously [...] So there are moments of discovery that I don't think I would come across if I was just picking any one of these to look at at a time."*

Participants also suggested several other areas for improvement: U2 noted that captions are mostly useful for refining sections found in the script, but less useful in isolation. All three users requested the ability to bookmark scenes to play back later. U1 and U2 also suggested showing some visualizations of aggregate statistics to inspire new questions and searches. For instance, U1 recommended showing a preview of all locations or characters in a movie could be helpful, whereas U2 wanted a list of frequently used verbs. U1 and U2 found the confidence bars helpful when deciding which results to click on, and U2 suggested filtering out results with low certainty.

5.9 Evaluation: Searching and browsing with SceneSkim

We used SceneSkim to conduct each type of query mentioned in our interviews with film studies researchers and film professionals. In order to identify realistic queries that might arise in the course of their work, we retrieved specific examples from existing film studies literature about the original *Star Wars* trilogy: *Star Wars Episode IV: A New Hope*, *Star Wars Episode V: The Empire Strikes Back*, and *Star Wars Episode VI: Return of the Jedi*. We instrumented our system to record interaction statistics and completion time while performing each task (Table 6.1).

Searching for characters: The essay *The Empire Strikes Back: Deeper and Darker* [32] makes the claim that the character Darth Vader becomes more menacing in *Episode V* than he was in *Episode IV*. To explore this hypothesis, we searched for "Vader" in the **summary** across both films. We clicked through the **summary** search results in *Episode IV* to view corresponding important events where Vader occurs. One summary result describes Vader dueling with another character. Clicking on this sentence navigates into the vicinity of the duel. We refine navigation by clicking on an explanation of the duel in the script. Another summary result describes Vader spiraling away in a spaceship. When we click this result, we see Vader spiraling away in a spaceship while yelling "What??" Comparing the results from both movies, *Episode V* depicts Vader in darker lighting with a louder wheezing noise, suggesting that Vader is portrayed in a more menacing fashion in *Episode V*.

Searching for objects: In *Your Father's Lightsaber* [121], Wetman notes that the screen time devoted to the "lightsaber," a futuristic weapon, increases throughout the first trilogy. To investigate this hypothesis, we searched for "lightsaber" in **script actions**, **summary**, **captions** and **script dialogue** across all three movies. Using search results from all three documents, we located all instances of lightsabers in the movie, watched the corresponding clips, and timed the amount

of screentime during which lightsabers appeared on screen. We found that scenes with lightsabers did increase through the three movies with 157s of screen time in *Episode IV*, 217s of screen time in *Episode V* and 258s in *Episode VI*.

Searching for dialogue: *Stoicism in the Stars* [45] uses 22 quotes from the first trilogy throughout the essay. We were able to locate and watch all 22 quotes using our system. We found the quotes by searching for quote terms in **captions** and browsing within the captions and scripts when multiple quotes were close to one another.

In the *Star Wars Episodes 4-6 Guide: Themes, Motifs, and Symbols* [3], the author notes that Luke's costumes change from white in *Episode IV*, to grey in *Episode V*, to black in *Episode VI*. Searching for Luke's appearances in **summary** quickly reveals this transition and several other variations of Luke's costumes (e.g., a pilot suit, a brown cloak, a Storm Troopers suit and a tan jacket). Brode and Deyneka [32] describe Hoth, Dagobah, and Cloud City in detail arguing that these locations represent the phases of Luke's journey in *Episode V*. We quickly locate and confirm these descriptions by searching **script locations** for "Hoth", "Dagobah" and "Cloud City". Based on Stephen's essay on stoicism [45] which suggests Jedi's encourage patience while the Dark Side encourages anger, we compare uses of the terms "patience" and "anger" in **captions** across the first trilogy to find that characters associated with the "dark side" explicitly encourage anger 3 times while Jedis discourage it 3 times (both encourage patience). Kaufman's essay [32] suggests that the robot character R2D2 shows affection towards other characters. To examine how R2D2 conveys affection without speech, we search for "R2D2 AND beeps" in **script actions** to find that the robot character shows emotions by beeping in a wide variety of pitches, tones, and durations. Finally, according to Gordon's [32] observation that there are "red lights flashing in the background whenever [Han] Solo and Leia confront each other" we search "Han AND Leia" in **scenes** finding flashing red lights do not occur during at least 4 conversations between Han and Leia.

5.10 Limitations and future work

Our current implementation of SceneSkim has both technical and practical limitations.

Availability of scripts and movies

The availability of scripts currently restricts which movies we can search over in SceneSkim, though we found informally that scripts are increasingly available for new movies. Captions and summaries are more readily available for both older and newer movies. In the future, movie library search may become available through sites like Netflix, which already have a paid usage model, or through micropayment schemes. In fact, video players like Amazon's TV X-Ray[3] already show metadata such as actor names on demand. In the present, individuals will have to load their own collections into our system.

[3] http://www.wired.com/2015/04/amazon-xray-fire-tv/

		Search result clicks			Document clicks				
Label	Task	Summary	Script	Captions	Summary	Script	Captions	Video watched	Completion time
A	Vader	10	0	0	0	3	0	3:48	5:37
B	lightsabers	1	22	1	0	37	0	20:00	21:44
C	Luke's costumes	17	0	0	0	16	0	4:23	5:20
D	22 quotes	0	2	19	0	0	5	4:31	8:36
E	main locations	3	24	0	0	2	0	2:57	4:23
F	anger/patience	0	0	12	0	0	0	00:21	1:37
G	R2D2 beeps	0	13	0	0	3	0	1:23	1:56
H	Han and Leia	0	16	0	0	5	0	2:28	3:10

Table 5.2: We instrumented our interface to record interactions while answering queries. "Search result clicks" refers to the origin document for search results we clicked in the search pane, while "document clicks" refers to clicks within the movie pane. While answering queries, we typically first clicked a search result from the summary, script, or captions then used the script or captions for fine grained navigation around that location. On average, we watched 5 minutes and 9 seconds of video for each query and spent a total of 6 minutes and 17 seconds completing the task.

Summary to script alignment

Our informal evaluation suggests that the alignment of summary sentences to script sections is useful high-level navigation, but the technique could be improved substantially. First, our approach does not yet support re-orderings of scenes between script and final movie, which can occur during editing. Summaries can also present events in a slightly different order than the source film, especially when the film contains scenes in which two or more narrative strands are interleaved. In addition, our method does not support a summary sentence that should match multiple non-contiguous script sections. Finally, our manually-constructed ground truth alignments are time-consuming to create (about 1 hour for every 20 summary sentences). Given a larger corpus of ground truth data, we could consider using a machine learning approach for summary alignment instead.

Adding new types of meta data

Some film studies researchers study shot length, lighting, set design, and shot types in order to analyze patterns in videos. Future work could employ computer vision techniques to detect visual attributes to allow users to search over more types of metadata. For example, users could study how a director used a particular shot type to convey a certain concepts by searching for that shot type in all movies by that director.

Adding more visualization capabilities

Currently, we visualize search results by displaying a text snippet and the result type. This visualization supports the tasks outlined in early interviews. However, our interface may reveal more patterns if we added visualizations for frequency of search results over different facets, such as time within the movie, genre, release date, writer, or director.

Adding bookmarks and correcting mistakes

In the informal evaluation, several users pointed out that they would like support for bookmarking, categorizing, and taking notes on video clips in order to keep track of many clips for research or teaching. Future work could add capabilities to support these common user tasks. In addition, since the algorithms do not achieve 100% accuracy, we hope to add tools within the system to correct algorithmic mistakes.

5.11 Conclusion

SceneSkim, a new interface forsearching and browsing videos through aligned captions, scripts, and summaries. Our system allows us to quicklysearch and browse clips in order to answer queries drawnfrom from existing film studies analysis. In the informal evaluation, three film studies researchers answered new questions using our tool, and enthusiastically stated they would use this tool for their work. While SceneSkim supports a domain where the use of video is native to the task, our next project moves to a domain where video is not typically used.

Chapter 6

VidCrit

6.1 Preamble

VidCrit lets users summarize casually recorded speech into a useful set of feedback for review. VidCrit achieves efficiency through turing the recorded speech into a navigable to do list while preserving the original content and structure. While VidCrit automatically segments and labels critiques, VidCrit gains flexibility by allowing the feedback recipient to edit and summarize critiques and allowing the recipient multiple options for browsing the feedback (e.g., by shot, by chronological order in original video, by chronoloogical order in critiques).

6.2 Introduction

Video review is a key step of the video production pipeline in which stakeholders provide feedback on drafts of video projects. The reviewer's feedback can include comments that indicate problems in the video (e.g., "the text is too small"), offer suggestions (e.g., "crop the shot") or give compliments (e.g., "I like the lighting in this shot"). In addition to commenting on such local issues, reviewers also give global critiques that pertain to the entire video (e.g., "In general, the music is too loud compared to the speech."). Video authors interpret and incorporate such feedback into subsequent drafts to improve the video.

In our formative study, we found smaller teams working on video projects typically give feedback informally, either verbally in-person or in asynchronously written text comments. Reviewers often prefer in-person feedback, because they can communicate their feedback more efficiently with speech than they can using text. For example, reviewers can easily communicate *temporal changes* (e.g. "move this shot here") by timing their spoken comments to events in the video and scrubbing the timeline to the locations they wish to change. Reviewers can indicate *spatial changes* (e.g. "move this text over to here") by gesturing over the source video with the mouse. In contrast, writing down temporal and spatial critiques can be tedious and time consuming, because reviewers first need to identify and transcribe timestamps, and then carefully describe proposed changes using text.

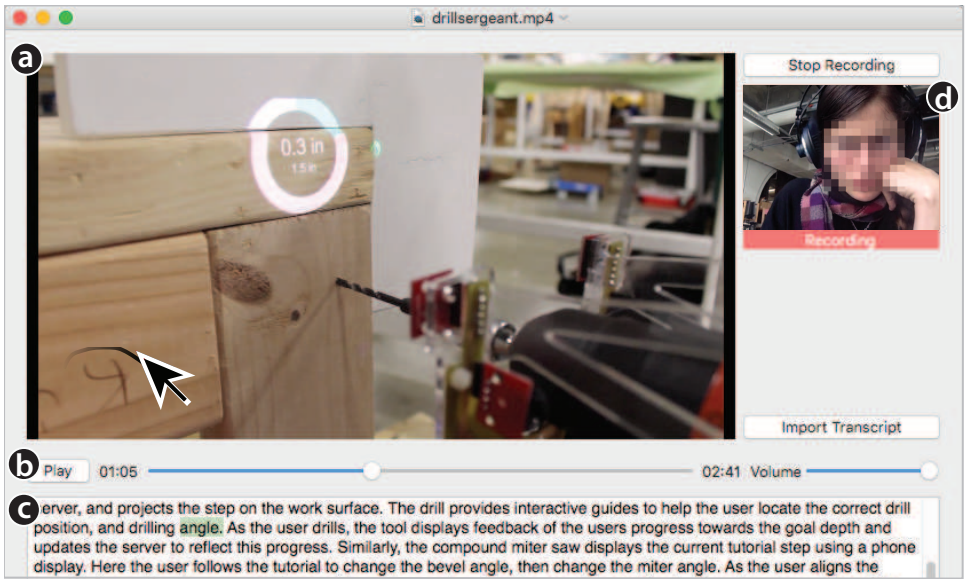

Figure 6.1: The feedback recording interface displays a video player for the source video (A) that the reviewers can play, pause, and scrub using the video timeline (B). Reviewers can mouse over the video to leave annotations (C). Reviewers can also import a source video transcript (D) using the "Import Transcript" button or start/stop recording using the "Start/Stop Recording" button. The recording interface also features a webcam camera view (E).

Video authors also prefer in-person critiques, because the context of comments can be more immediately understood. Authors can observe the reviewer's tone of voice and other non-verbal cues to better interpret verbal comments. For example, the comment "Come on, what is this mess?" could either be a joke or indicate a serious problem, depending on the reviewer's tone of voice. They can also engage in a dialog with the reviewer to come to a shared understanding of requested changes. However, written comments have the advantage of providing the author with a documented list of critiques. We interviewed four video authors who estimate that, even though they prefer in-person feedback, 80% of feedback occurs over e-mail because of scheduling and location constraints.

In this paper, we investigate how to preserve favorable qualities of both in-person and written text critiques, while allowing the feedback to occur asynchronously. By conducting formative interviews with practitioners, we develop a set of design guidelines for giving and receiving feedback. We then present VidCrit, a system that incorporates these guidelines into two interfaces; (1) a feedback *recording interface* (Figure 6.4) that lets reviewers efficiently capture their feedback, and (2) a feedback *viewing interface* (Figure 6.6) that lets video authors view this feedback later. The feedback recording interface captures the reviewer's spoken comments, facial expressions and hand gestures (using a video camera), the current playback time in the source video, interactions with the source video timeline (e.g. scrubbing, play, pause), and mouse gestures over the video player and transcript.

Our feedback viewing interface allows the video author to browse, skim and edit the feedback

provided by the reviewer. The interface links the audio track of the reviewer's speech with the time-point the reviewer was watching in the original video, so that the video author can view the reviewer's comments in context. Authors can navigate these context-based comments using a timeline, or by browsing and skimming an automatically segmented list of transcribed comments. We automatically segment comments using a novel technique which achieves precision and recall of 0.836 and 0.835 respectively. We also automatically label each transcribed comment with several comment properties to help authors choose which comments to view. For instance, we mark which comments pertain to a short portion of the video (local comments) or the video as a whole (global comments).

We conducted a user study and found that, when reviewing 3 minute videos, reviewers provided 10.9 ($\sigma = 5.09$) more local comments using our interface than when using text. In a Likert-scale survey users compared the recording interface (5) to text (1) on a number of attributes. Overall, users preferred our interface ($\mu = 4.6, \sigma = 0.48$) for providing comments. We also asked both amateur and professional video authors to create a to-do list of edits using our feedback viewing interface. All video authors found our interface preferable to receiving feedback via e-mail.

In summary, contributions of VidCrit include:

- A set of design guidelines for our interfaces for giving and receiving feedback derived from current practice,
- the VidCrit recording and viewing interfaces for communicating and receiving feedback asynchronously,
- techniques for segmenting and labeling comments, and
- a user study comparing feedback provided with our recording interface to feedback provided using text, and an informal evaluation of our viewing interface.

6.3 Prior Work

Our VidCrit system allows video reviewers and authors to communicate feedback. It builds on prior work in three areas; (1) video logging and annotation, (2) feedback interfaces that support text comments, and (3) feedback interfaces that support speech comments.

Video annotation tools

Recording comments about a source video is closely related to video logging, the process in which practitioners watch and label a video according to its content. Video logging helps practitioners categorize video clips for later review, and cut down the amount of video they will need to re-watch. Early systems for logging videos addressed individually logging clips during the recording session [85], after the recording session [140], or as a group after the recording session [42, 43, 18]. Later work improved video logging usability and efficiency by creating fluid tablet interactions [113], allowing users to apply tags to keyframes [139], accommodating voice

annotations [133], and segmenting videos at the per frame level [47]. Whereas previous work mainly addresses generalized tasks of assigning tags and writing notes on video clips, we support the specific task of video review, helping reviewers produce specialized annotations. Using the spoken word, reviewers can express nuances that are hard to convey in written text. By recording the position in the video as the user scrubs, our system allows reviewers to indicate points in the video timeline, without using timecodes. And by recording the mouse position, our system allows reviewers to indicate spatial regions of a video frame using mouse gestures instead of textual descriptions. For video authors, we support searching and browsing the feedback session using a variety of interactions.

Text-based asynchronous feedback interfaces

Marqueed [10] allows users to provide feedback on images via annotation tools and text, and provides support for discussion on each comment. Frame.io [6], ScreenLight [12] and Wipster [17] build on this idea but allow reviewers to draw and provide text comments on any single frame of a video. Users can navigate the source video by clicking a reviewer's text comments on the side of the player. However, because these interfaces link each comment to a single frame, they do not easily support delivering comments about a range of frames or about multiple discontinuous time ranges. Unlike VidCrit, these prior systems do not accommodate any spoken feedback.

Speech-based asynchronous feedback interfaces

UserTesting.com [16] and Silverback [13] are designed for practitioners conducting user studies on websites or software to record study sessions. These tools record a screencast of the task, the voice of the user, and the user's webcam video (Silverback [13] only) and present these feeds as one video segmented by task. VidCrit builds on this approach by transcribing and segmenting a reviewer's feedback by topical comments, allowing video authors to easily skim and browse comments. In the context of communicating feedback on PDF documents asynchronously, Yoon et al.'s RichReview [142] and RichReview++ [143] support voice comments supplemented with mouse gestures. The task of video review faces different design challenges than PDF review: Whereas RichReview implements spatial annotations via drawing, our system also records and replays temporal annotations such as scrubbing in the video. In addition, we consider how to allow users to view such temporal annotations and gain context for diectic comments by navigating synchronized source and feedback video timelines.

Cattelan et al. [36] also allow users to leave speech comments on a video so that non-collocated viewers can watch a TV show together. One viewer can leave comments on the video timeline while pausing the video, and another can only play back the comments in order by watching the video. Unlike Cattelan et al., our feedback recording interface lets video reviewers leave comments continously without pressing record and our feedback viewing interface lets video authors index those comments without watching the entire video.

Finally, FrameBench [5] lets reviewers and authors remotely collaborate synchronously using a shared video player. Due to scheduling constraints and bandwidth limitations, synchronous collaboration isn't always practical, so our work focuses on asynchronous collaboration instead.

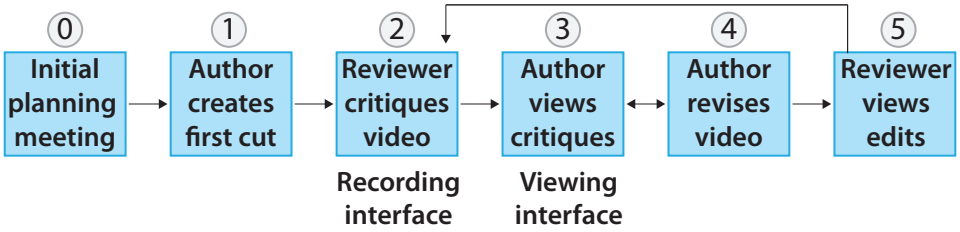

Figure 6.2: A typical editing workflow proceeds as follows: First, (0) the team meets synchronously to talk about the plan for the video and gathers the footage before or after this meeting. After this, (1) the video author (e.g. editor, producer/editor) creates the first cut, (2) a reviewer (e.g., client or executive producer) provides feedback, (3) the author views critiques, then (4) revises the video, and (5) the reviewer views the edits. The team then repeats steps 2-5 until both parties find the result satisfactory, or they reach a predetermined number of iterations.

6.4 Current Practice

Video workflows are well studied in the context of sharing and watching home videos [76], creating TV episodes [24, 23], and individuals editing videos [38, 27, 113]. However, no prior work investigates the process of small teams giving and receiving feedback for editing videos. To guide the design of our system for communicating such feedback during the editing process we asked practitioners: What are the benefits and drawbacks of the methods video editors currently use to communicate feedback?

We conducted semi-structured interviews with four video authors – three professionals and one amateur. These participants included an owner and producer for a small production company, a producer for a local TV news station, a video producer on staff for a university campus, and a university student who has created several research videos. We asked participants to describe their production workflow and compare experiences with asynchronous and synchronous feedback. All participants had received both types of feedback.

We found video editors and producers follow a common editing workflow (Figure 6.2). Although producers preferred to provide and receive such feedback in person, 80% of feedback occurs asynchronously through e-mail with text comments and timestamps. Note that although the practitioners we interviewed use text when communicating asynchronously, and voice when communicating synchronously, there exist methods for communicating voice feedback asynchronously (e.g. Watch-and-comment [36]), or for communicating text feedback synchronously (e.g. instant messaging). However, we compare the pros and cons of asynchronous text communication (e-mail review) and synchronous voice communication (in-person review).

Interviewees mentioned several benefits of in-person reviews:

- **Efficiency for the reviewer:** Composing a succinct and complete list of e-mail comments takes more time than watching and describing the changes in person.
- **Discussion and brainstorming:** In person, the author and reviewer discuss changes in the case of disagreements, and brainstorm alternatives to problems.

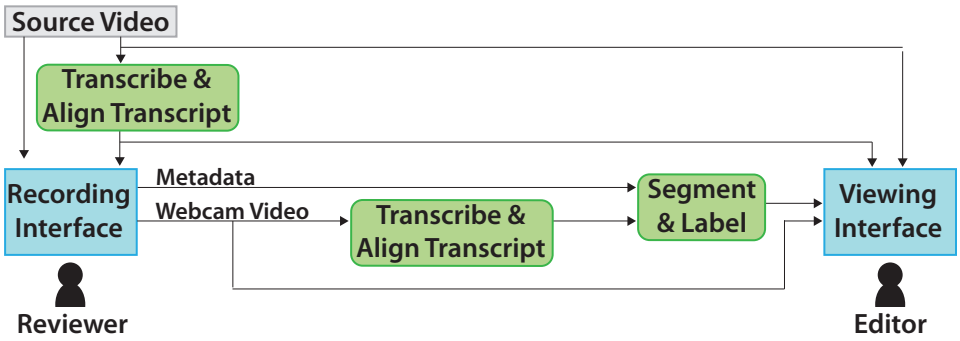

Figure 6.3: VidCrit takes a source video – and optionally, a source script – as input. If the reviewer does not provide a script, the system transcribes the video using rev.com [11], a crowd-based transcription service. VidCrit aligns the transcript or script to the source video. Then the reviewer uses the recording interface to record their feedback on the video, and the system transcribes and aligns the reviewer's webcam video. The system segments the session into comments by considering the interaction metadata, the transcript, and the aligned transcript timestamps. The author reviews the segmented feedback session and synced source video using the viewing interface.

- **Body language and tone of voice**: Authors benefit from observing body language and tone of voice, as these convey affect of the reviewer's comments and initial reactions to the video draft. One author mentioned that he tries to "read between the lines" of text comments, because he cannot tell if the video works for the client through an e-mail list of critiques alone.
- **Additional comments**: Authors mentioned they received additional comments in person, including off-the-cuff reactions and positive comments, that may not be conveyed in an e-mailed list of changes.
- **Trying alternatives:** Co-located editors and producers sometimes try alternative edits in person to reduce the number of iterations.

Interviewees also mentioned a few benefits of e-mail reviews:

- **Efficiency for the author:** E-mail comments from the reviewer provide an explicit to-do list of edits for the video author.
- **Accountability:** When the author finishes the list of documented changes, the author has completed their job and can resolve disputes by pointing to recorded e-mail comments. No such record exists for in-person reviews.

Through our interviews with practitioners, we identified a set of benefits of both asynchronous and synchronous feedback. Because synchronous feedback is often unfeasible, we build a system for asynchronous feedback that preserves many of the identified benefits. As we are focusing on building an asynchronous system, we do not provide features for synchronous discussion, brainstorming, and trying alternatives. Instead, we consider how to keep the process efficient for both

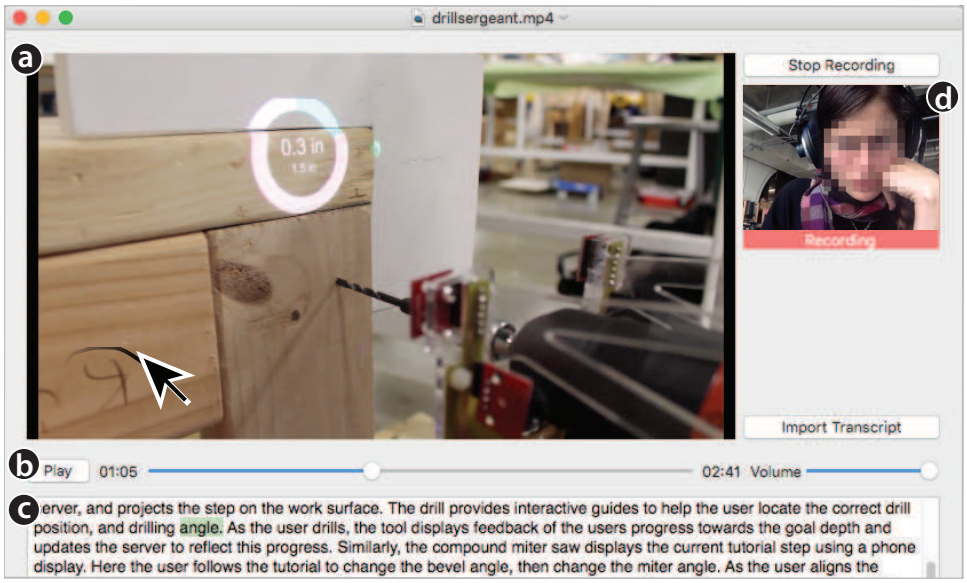

Figure 6.4: The feedback recording interface displays a video player for the source video (A) that the reviewers can play, pause, and scrub using the video timeline (B). Reviewers can mouse over the video to leave annotations (C). Reviewers can also import a source video transcript (D) using the "Import Transcript" button or start/stop recording using the "Start/Stop Recording" button. The recording interface also features a webcam camera view (E).

the reviewer and the video author. In addition, we seek to let authors observe the reviewer's body language, tone of voice, and hear additional comments as they could in person.

6.5 Interfaces

VidCrit (Figure 6.3) features two interfaces: one interface for reviewers to record their critiques on a video (the feedback *recording interface*) and another interface for video authors to view the critiques (the feedback *viewing interface*).

Feedback Recording interface

The recording interface (Figure 6.4) lets reviewers capture spoken feedback on a source video. The reviewer watches the video in the interface and delivers spoken critiques about the video much as they would in person. The interface records the reviewer's speech and their playhead location at all times, in order to capture the context of each comment. The interface also records the reviewer's webcam video (Figure 6.4E) so that the reviewer can supplement critiques with facial expressions and hand gestures.

The reviewer can pause, play, and seek within the source video using the timeline (Figure 6.4B) to easily record temporal critiques. For example, the reviewer may watch the video until she finds a problem such as an incorrect ordering of two video segments. The reviewer can pause the video

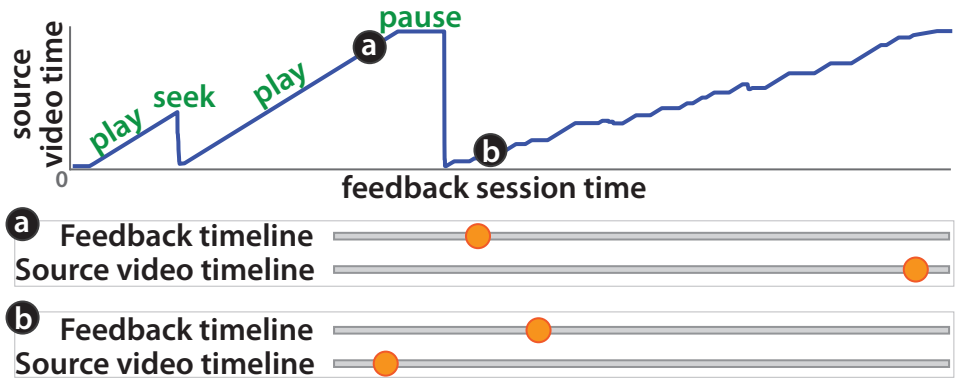

Figure 6.5: The source video time vs. feedback session time chart displays the reviewer's location in the source video during the feedback session. Here, the user plays the first part of the video, then seeks to the beginning to replay the whole video. Then, the user pauses the video and seeks to the beginning of the video. After this, the user slowly progresses through the video, playing, pausing, and seeking while giving comments. The timelines (A) and (B) show how the position of the source video timeline in the viewing interface updates depending on position of the feedback session timeline, according to where the reviewer positioned the source video at each time.

and say "Move this part from here, back to here" while scrubbing on the timeline to indicate the relevant locations in the video. The reviewer can also draw on the video player (Figure 6.4C) to describe spatial feedback.

For instance, if the reviewer notices distracting markings on a piece of wood, she can pause the video and say "We should remove the extra marks on the wood." while gesturing over the video player with the mouse, or scrubbing to other parts of the video to show additional points where distracting marks occur. Alternatively, the reviewer can include reactions as the video plays such as commenting "These are good images." while viewing a shot she likes.

The reviewer can optionally import a source video transcript (Figure 6.4D), which the system aligns to the video such that the transcript highlights the currently spoken word as the video progresses, as in prior work [27, 106]. Reviewers can use this transcript to navigate the source video by clicking on a word, and to provide feedback on wording or grammar in the source video script. For example, a reviewer can provide feedback by saying "I think instead of 'multiple tool tutorials' we should say, 'tutorials which leverage multiple tools'" while highlighting the corresponding section of the source video transcript.

Feedback Viewing Interface

After a reviewer finishes recording feedback with the recording interface, the video author can open a review session in the feedback viewing interface (Figure 6.6). The interface consists of two panes: (1) the *direct navigation pane* lets video authors watch and navigate the synchronized feedback session and source video, and (2) the *segmented comments pane* lets authors browse the feedback by segmented text critiques.

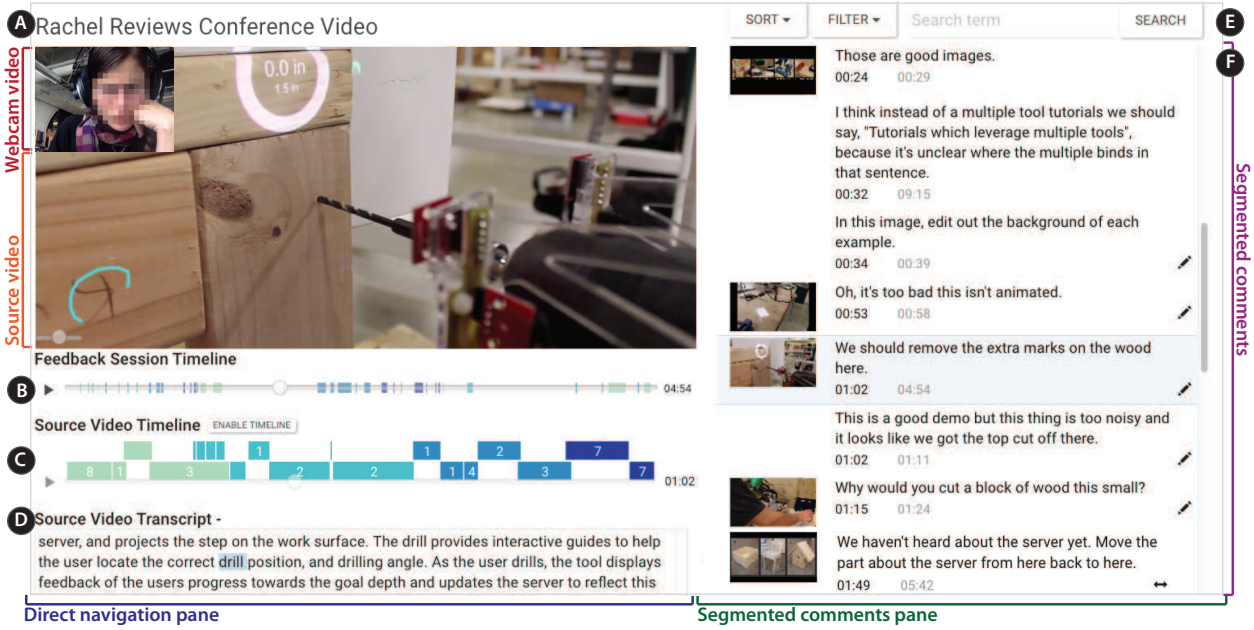

Figure 6.6: The VidCrit interface consists of a **direct navigation pane** for navigating the feedback session using the webcam and source video timelines, and a **segmented comments pane** for reviewing transcribed and segmented critiques. The direct navigation pane features the source and webcam videos along with a title (A), a feedback session timeline (B), a source video timeline (C) and the source video transcript (D). The segmented comments pane features sorting, filtering and search options (E), along with a list of segmented comments (F).

When reviewing feedback, the video author now has two pieces of time-based media to negotiate: In addition to the temporal order of the original source video under review, the feedback itself has its own temporal order, which may not relate to the original source video temporal order in any simple way (Figure 6.5). At one point, the video author may want to navigate using the temporal order of the source video, for example to address multiple comments about a single shot that may have occurred at different times in the feedback video. At another point, the video author may want to navigate using the temporal order of the feedback video, for example to understand a sequence of related comments that refer to different shots in the source video. The feedback viewing interface therefore features two timelines: the *feedback session timeline* and the *source video timeline*. Either timeline can be used to navigate, and navigation using these timelines can be linked or unlinked as described later in this section.

Direct navigation pane

The direct navigation pane shows the feedback session webcam video and the source video (Figure 6.6A). The author navigates the feedback session by playing, pausing, and scrubbing with the feedback session timeline (Figure 6.6B). When the author plays the feedback session, the reviewer's webcam video and speech comments advance linearly, while the source video and the

source video timeline (Figure 6.6C) update to reflect the reviewer's position in the source video during the feedback session. For example, if, during the feedback session, the reviewer paused the source video to say "We should remove the extra marks on the wood", the source video will later pause at the same point when the video author replays the feedback session. Figure 6.5 shows an example how the synchronized timelines update as the video author plays back a feedback session.

Because the video reviewer often remains silent as they watch the video, a large portion of the feedback session does not contain feedback. To let the video author avoid silent portions of the feedback session, the feedback session timeline shows the duration of reviewer comments with colored rectangles. The color of these colored rectangles represents whether the comment addresses the beginning (light green), middle, or end (dark blue) of the source video (Figure 6.6B).

Video authors can see how the reviewer moved through the video by glancing at the comment colors. In Figure 6.6B, the reviewer watched the video two times while making comments then skipped around the video to make additional comments.

Video authors may also unlink these timelines with the "Unlink timelines" button and then navigate (e.g., play, pause or scrub) within the source video independently, while the feedback session video, audio and timeline remain paused. For instance, if the video author hears the reviewer say "Why would you cut a block of wood this small?" the author may need to rewatch the corresponding source video clip to gain context. The video author may view the section with the block of wood a second time by clicking "Unlink timelines" (Figure 6.6C), scrubbing back to the relevant location, then pressing play. Afterwards, the author can click "Link timelines" to continue browsing the synchronized feedback session and source video.

As in prior work [27, 106], the source video transcript (Figure 6.6D) highlights the currently spoken word as the source video progresses, enabling authors to gain context for comments. In addition, the source transcript displays the reviewer's transcript selections from the feedback session.

Segmented comments pane

The segmented comments pane allows authors to navigate the feedback session using transcribed and automatically segmented comments (Figure 6.6F) that each pertain to a single-issue critique. Each comment corresponds to a certain time in the feedback session (*feedback session time*) and a certain time in the reviewed source video (*source video time*). The *feedback session time* (Figure 6.6G) is the starting time of the first word in the comment, and the *source video time* (Figure 6.6H) is the source video playhead location at the *feedback session time*. Clicking on a comment navigates to the corresponding times in the synchronized feedback session and source video. Pressing 'space' plays and pauses the synchronized feedback session and source video.

The shot-based timeline (Figure 6.6C) features colored bars that each represent a shot.

We construct the shot-based timeline using the Edit Decision List (EDL) exported from the video author's editing software (Figure 6.7). The EDL file, a standard format exported from common video editors (e.g., Adobe Premiere, FinalCut), gives the start and end time of each shot within the final edit along with the filename of the footage. To build the shot-based source video timeline, editors import one EDL per video track in the project and we parse the EDL files to find each shot's in and out time, and filter out all audio or blank shots. As some video editing programs do not support EDL export, we also supply automatic shot detection at as a fallback [56, 88]. We

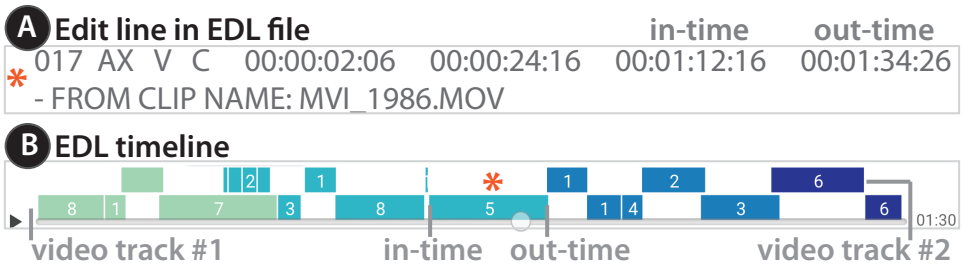

Figure 6.7: Each line in an EDL file (A) specifies a clip's properties along with its filename and in and out point in the final video file. Each segment in the EDL timeline (B) represents one line in an EDL file.

include further discussion of the automatic shot detection in the Algorithms section. Authors can navigate to all comments about a shot of the source video by clicking on it in the shot-based timeline. This is useful when reviewers make several passes through the source video. For example, if the last shot in the video is less compelling, the video author may want to first view all critiques about that part of the video. The numbers in each colored bar (Figure 6.6C) represent the number of comments attributed to that shot. For instance, the last bar shows there are 7 comments on the last shot.

We show one thumbnail per shot, grouping together consecutive comments that occur within the same source video shot. We show only one thumbnail per shot to avoid repetitive thumbnails.

The comments (Figure 6.6F) also show icons that provide additional information about each comment: a pencil shows the comment contains a mouse gesture, the arrows show that the reviewer scrubbed to multiple times during the comment, and the globe shows that the comment pertains to the video as a whole rather than one specific location. As authors skim the segmented comments, the comment text and corresponding source video frame may be sufficient to understand some comments. A scrubbing or annotation icon indicates that the reviewer conveyed additional information that the author can only view by playing the comment.

As the author reviews the comments, she may hide irrelevant comments such as "Hmm, lets see." by deleting the comment, or hide comments she has already addressed by marking them as completed. When the video author hides a comment, the interface removes the corresponding mark on the feedback session timeline (Figure 6.6B) and decreases the comment count on the EDL timeline (Figure 6.7). Finally, authors can edit the text of long comments by entering edit mode (command-alt-click on a comment) (Figure 6.8). For example, an author might edit the second comment in Figure 6.6F down to *"Replace 'multiple tool tutorials' with 'tutorials which leverage multiple tools' "*, to make the comment easier to read on subsequent passes. In editing mode, the authors may also toggle the global/local icon (Figure 6.8A) or delete/complete a comment.

The author can sort, filter, and search through the segmented comments using the option bar (Figure 6.6E). The sort feature lets the author sort comments in various ways. Sorting by the feedback session time enable the author to see the comments chronologically. Sorting by the source video time allows her to see the comments grouped by shot in the source video (displayed). Or sorting by comment duration makes it possible to see the longest comments first, as they may

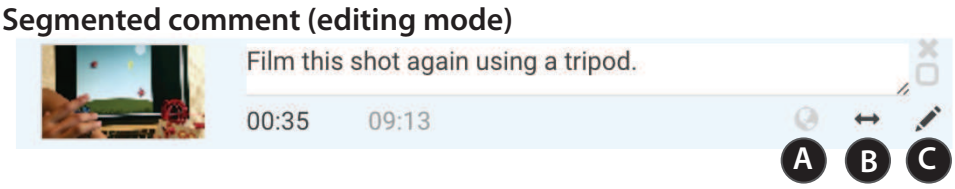

Figure 6.8: This figure shows a single comment in editing mode with three optional icons (A,B,C). The global icon (A) displays whether a comment relates to the entire video or just the given location. The scrubbing and annotation icons (B,C) display whether a comment has a corresponding scrubbing or annotation action.

take longer to understand and address than the shorter comments. The filter feature lets the author view comments that fit any one of the comment icon properties. For instance, the video author may show or hide all global comments. The search feature lets authors search over the comment text, and the source video text (e.g. comments that pertain to a subsection in the source video where they say "system").

6.6 Algorithms

Our system transcribes the source video and aligns the transcript to the video to enable transcript-based interactions. The system also transcribes, aligns, and segments the feedback session into single-issue comments so that video authors can browse and search the feedback session using text. We automatically assign three labels to each comment, and choose a preview thumbnail to provide authors with more information. Finally, we segment the source video into shots using Edit Decision Lists and automatic shot detection.

Transcribing and aligning videos

Aligned transcripts of the source and feedback videos are useful for reviewers and authors to skim for content of interest, index the video, and gain context for comments. Since video creators often write scripts before shooting and editing source videos, we use the original script where available. To transcribe feedback videos and unscripted source videos, we use rev.com [11] – a crowd-based transcription service that charges about $1 per minute of audio. Following prior work, we align the transcript to the source video by concatenating all text and the corresponding audio segments that are part of continuous speech [103]. We align each segment of continuous speech with its corresponding audio segment, or an existing script with the entire audio track, using forced alignment between phonemes and audio features, as in the work of Rubin et al. [118]. Reviewers can then navigate using the transcript and annotate the transcript (Figure 6.4), while video authors can search and browse reviewer comments using text (Figure 6.6F).

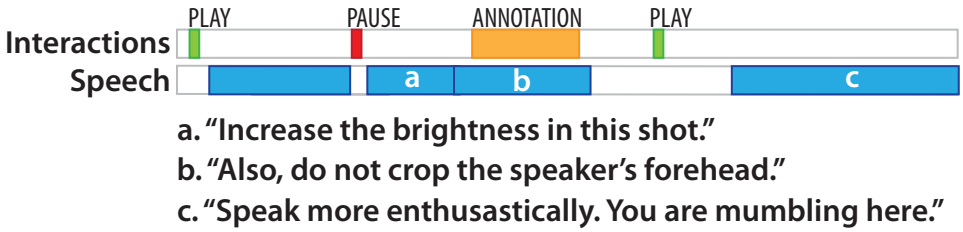

Figure 6.9: Comment (A) appears right after the reviewer pauses the video, indicating that they might begin a new comment. As the comment (B) begins with the word "also" and an annotation occurs slightly following the beginning it also may be a comment. Finally, comment (C) occurs after a long break between speech, so it may be the beginning of a new thought.

Segmenting comments

After transcribing and aligning the feedback session transcript, our system segments the feedback session into topical comments so that the video author can quickly skim and browse the feedback. We use the text transcript, the alignment timestamps and the interaction metadata from the feedback session as inputs to the segmentation algorithm.

Segmentation Algorithm

Our system transcribes comments into single issues so that authors can manipulate, browse, and play back comments in issue-based chunks, much as they would items in a to-do list. In the dataset we collected, almost all sentences contain a single issue (a few contain more than one issue). So our system first generates possible issue segment boundaries as the beginning time for the first word in each sentence. Since our system's punctuation comes from a crowd transcription site, some transcribers may not include a period if a user trails off without finishing a thought. Therefore, our system adds all words that occur after a pause in speech as potential segment boundaries.

Given a set of potential segment boundaries, we classify whether each one is true issue boundary using an SVM in combination with a Random Forest Classifier. The classification relies on the video reviewer's interactions recorded during the review session, and the aligned transcript of the feedback video. Our system calculates the following feature sets for each potential segment boundary to help determine whether it is a true segment boundary:

- *Interaction proximity*: Proximity to each type of interaction (pause, play, seek, gesture) before and after the potential segment boundary
- *Word proximity*: Proximity to spoken words immediately before and after the potential segment boundary
- *Segment length*: Includes segment duration (time from beginning of first word to end of last word), and the word count of the segment
- *First word*: The first word of the potential segment

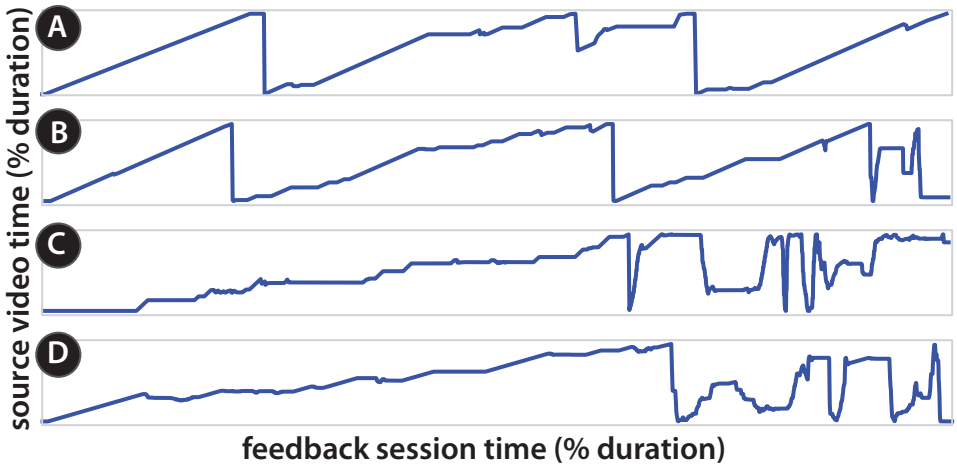

Figure 6.10: This figure shows the reviewer's time in the source video vs the time in the feedback session for 4 different reviewers (A,B,C,D). In the first examples (A and B), reviewers watch the video multiple times while playing, pausing, and scrubbing to leave comments after the first time. In the second examples (C and D), reviewers pause and scrub to leave comments on their first pass. At the end of the session, they give general comments while scrubbing through the video to find examples.

- *Punctuation*: The punctuation of the prior segment

We chose these features by observing different ways that reviewers conversationally segment their comments. For example, we include interaction proximity because reviewers often pause the video before they start a new comment, play after finishing a comment, and seek or gesture briefly after starting a comment to convey temporal or spatial critiques (Figure 6.9A). We include word proximity because reviewers often pause speaking between comments (Figure 6.9C). The segment length features help us identify very short potential segments such as "Okay.", "Yeah.", "Umm, let's see." or "Mmhmm," which often contain only verbalized pauses, and such standalone expressions represent new segments. We include the first word of the current segment because the first word may be a transition word that can be used to start a new thought (Figure 6.9B). Finally, we include punctuation because a pause (no punctuation) is more likely to indicate hesitation rather than finishing a thought.

We use a Linear SVM with L1 penalty, a common penalty for feature selection, to remove unimportant features (i.e. features with a coefficient of zero). The *first word* feature set generates one feature for each word that starts a potential segment. As a result, the feature selection step typically removes first word features that do not correspond to any positive examples. Then, we use a Random Forest Classifier with 3000 estimators and a maximum depth of 4 to train the classifier only using the selected features. The training and testing sets are described in the evaluation section below.

Video title	Subject	Time (min)	Feedback time (min)	Total comments	# Comments labelled		
					Annotation	Scrubbing	Global
Erin's famous fruit pizza [4]	cooking tutorial	2.87	11.98	36	4	23	4
LearnKendama.com kickstarter video [9]	product promotion	3.30	11.18	32	10	15	1
Interface demo video (unpublished)	interface walkthrough	2.73	13.52	34	13	4	0
Skintillates video [14]	project teaser video	1.00	10.38	29	14	12	1
Fabrication project video (unpublished)	project demonstration	2.68	13.57	60	26	9	2
France travel log [7]	travel	3.48	10.62	24	5	8	1

Table 6.1: Information for result videos. We segmented and labelled all videos automatically. Total comments refers to the number automatically produced segments. The number of comments containing an annotation, scrubbing, or global critique was also determined automatically.

Evaluation

To evaluate our comment segmentation technique, we collected short videos in a variety of domains, including a cooking tutorial, a Kickstarter video, two technology demonstration videos, and a travel video log which were an average of 3.01 minutes long ($\sigma = 19.5s$). We invited 8 people who had prior experience creating and giving feedback on video projects to provide feedback on one video each. Recorded feedback sessions averaged 12.4 minutes long ($\sigma = 1.61min$) where the subject spent an average of 3.49 minutes speaking ($\sigma = 42.4s$) and the remainder of the time listening and watching the source video or scrubbing through it. We manually segmented the resulting feedback sessions to create ground truth segmentations.

We observed reviewers utilizing various strategies for providing feedback on the videos (Figure 6.10). Some watched the video all the way through silently, then watched again while playing, pausing, scrubbing, and leaving comments multiple times (Figure 6.10A,B). Others paused and scrubbed during the first watch-through to add comments as they went, and at the end left general comments, scrubbing through the entire video to show examples (Figure 6.10C,D).

Using leave-one-out cross-validation with this dataset we found that our classifier identifies true boundaries with a precision, recall and F1-score of 0.836, 0.835, and 0.834 respectively (Table 6.2). In practice, a missed segmentation results in a longer comment segment to watch with little effect on the authors experience. A false positive segment boundary results in two segments that are topically the same. When leaving out individual feature sets, we find that leaving out word proximity results in the largest difference in F1-score. However, even without word proximity, the classifier performs much better than random (F1-score=0.633 vs F1-score=0.475).

Determining comment attributes

VidCrit assigns each comment three attributes with corresponding icons (Figure 6.8A-C) which let the editor know if the reviewer left additional information with the comment, or if the comment pertains to a local or video-wide critique. VidCrit also selects a thumbnail for each comment.

Global icon: We show the global icon (Figure 6.8A) if the words "in general", "as a whole", or "overall" appear in a segment as reviewers in our dataset used such language to describe general comments applying to the entire video. We created ground truth data for all of the videos in our

	Precision	Recall	F1-score
Ours	0.836	0.835	0.834
No interaction proximity	0.840	0.834	0.833
No word proximity	0.664	0.636	0.633
No segment length	0.829	0.827	0.827
No first word	0.831	0.829	0.827
No punctuation	0.836	0.833	0.832
Random	0.486	0.471	0.475

Table 6.2: Comparison between our segment boundary selection, leaving out feature sets, and randomly selecting boundaries. Random represents a function that randomly assigns a boundary or not weighted by the number of boundary occurrences in the training data.

dataset. This method achieves a much higher precision than recall for labelling global examples (0.80 and 0.33 respectively). That is, comments labelled "global" are likely correctly labelled, but this method misses many global comments. The method misses many such comments because global comments do not necessarily need to contain the specific words "in general", "as a whole" or "overall" to convey a comment about the entire video (e.g. "the audio quality is poor"). We would prefer to use a more robust n-gram classification based approach, but global comments are rare within the dataset we collected. In the future, we will collect more data to enable better classification.

Scrubbing icon: We show the scrubbing icon (Figure 6.8B) if the reviewer scrubbed within the video timeline at any point during the comment.

Annotation icon: The annotation icon (Figure 6.8C) displays whether the reviewer performed a spatial annotation within the comment. We record annotations as mouse movements over the video player, but many of these movements are unimportant or accidental. For example, users often mouse over the video player incidentally while navigating the timeline, which lies immediately below it.

To filter out such unintentional movements, our system first deletes any mouse event that occurs at the lower 10% of the video player pane. Next, it groups together continuous streams of mouse movements (less than 0.1 second apart) and computes the duration of each resulting segment. Our system shows the spatial annotation icon for any comment with a total mouse-movement segment duration over 0.3 seconds. We created ground truth data for three videos by labelling comments that contained meaningful annotations as true, annotation and comments that contained no annotations or accidental annotations as false annotations. Our approach achieves an annotation-labelling accuracy of 92%, whereas counting any mouse movement as a true annotation achieves an accuracy of 80%.

Thumbnail: We choose the thumbnail time as the start time of the first word in a comment unless a spatial annotation occurs. If such an annotation occurs, we choose the starting frame of

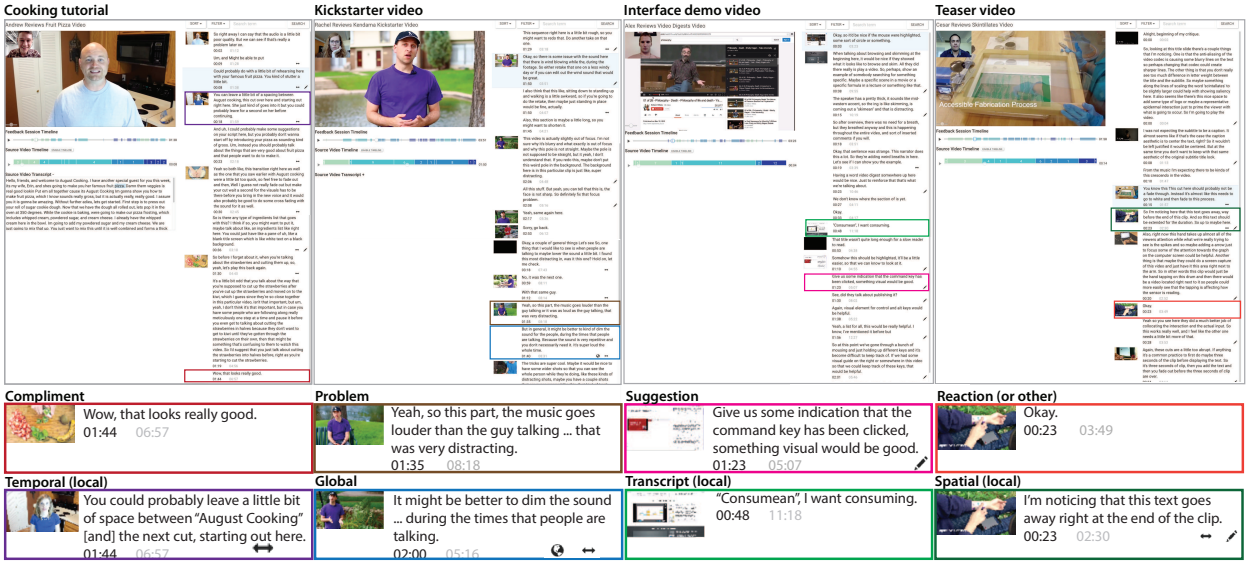

Figure 6.11: UseUsers critiqued four videos on a variety of topics. We transcribed these feedback sessions using rev.com [11], then our system automatically segmented the sessions into comments and labelled them. Users were able to produce a variety of critiques using our system including different critique styles (e.g. compliment, problem, suggestion) as well as critiques with different content (e.g. temporal, spatial, transcript and global).

the longest annotation. We hide the comment thumbnail if the comment follows another comment in the same shot. We describe our shot segmentation methods in the "Shot-based source video timeline" section.

Shot-based source video timeline

We provide a shot-based source video timeline, and comment grouping to allow authors to browse comments by source video section. We create this timeline based on edit decision lists (as described in Interfaces) when avaiable. When editors do not provide EDL files, VidCrit relies on a automatic shot detection based on color histograms to segment the source video into shots [56, 88]. In our implementation, we calculate color histograms for every frame, then compute the Chebyshev distance between each pair of histograms. We smooth the distance signal by convolving it with a Hanning filter, then identify large color histogram changes by finding all local maxima. However, not all local maxima correspond to shot changes as some may indicate smooth changes in lighting or camera angle. So, we find the local maxima with the sharpest peaks which indicate an abrupt color change by comparing each local maxima to the point before it, and only count the local maxima as a shot change if the difference is greater than 0.5 (we determined this threshold empirically based on the videos in our dataset).

6.7 Results

Figure 6.11 shows feedback sessions automatically segmented into critiques and labelled using our system. We generated these results using a set of videos in a variety of domains (see Table 6.1). We invited reviewers with experience creating videos and giving feedback on videos to use the recording interface to critique a video. To guide reviewer feedback, we gave the reviewers a set of goals for the video (e.g. convince people to financially back the project, get the general public excited about your research project). Reviewers spent an average of 11.88 ($\sigma = 1.28$) minutes delivering feedback, and produced an average of 36 ($\sigma = 11$) segmented comments. On average, 12 ($\sigma = 7$) comments contained scrubbing and 12 ($\sigma = 6$) comments contained an annotation, while only 1.5 ($\sigma = 1.3$) comments contained a global comment. These results suggest the reviewers used the annotation and scrubbing capabilities not afforded by text.

6.8 User evaluations

We conducted a user study with our recording interface and an informal evaluation with our viewing interface.

Recording interface user study

To find out if video reviewers communicate feedback more efficiently using our interface than using text, we conducted a user study with 8 participants who had prior experience both providing and receiving feedback on video projects. Our system focuses on asynchronous feedback, so we leave a comparison of our interface to synchronous, in-person feedback for future work.

Method

We selected two videos which contained content designed for a general audience (a Kickstarter video and a cooking tutorial), which were 3.32 minutes and 2.85 minutes long. We picked these videos because they represent different domains, production styles, and levels of formality.

We recruited 8 participants (6 males and 2 females, ages 23-31, graduate students) who had prior experience giving and receiving feedback on videos via school mailing lists. We allowed each participant 11 minutes to provide feedback on one video using text and 11 minutes to provide feedback on the other video using our recording interface. No participants had seen either video before the study. Between participants, we varied which video went with which feedback method, the order of the videos, and the order of the feedback methods. Two participants completed each of 4 possible orderings.

Before the participants provided feedback, we gave each participant the communication goals of each video, a list of types of feedback they might provide (e.g., spatial, temporal, transcript, overall), and a one minute overview of the recording interface. We concluded each study session with Likert scale questions and a semi-structured interview about advantages and disadvantages of each method.

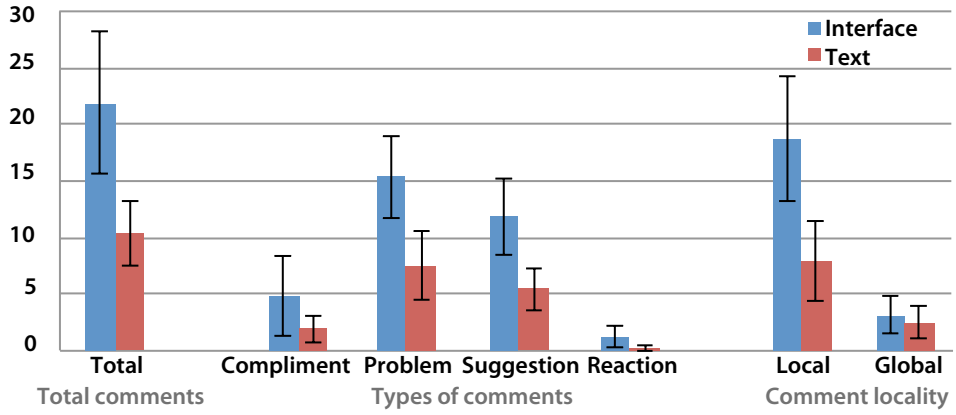

Figure 6.12: Average number of critiques produced using the interface and text total. Then average number of critiques produced using the interface and text which were assigned each label. Multiple labels may be assigned to one critique. Error bars show the 95

Recording Interface Study Results

We transcribed all recorded feedback sessions using the transcription service rev.com [11], and segmented the comments into single issues by hand. Because a comment may contain specific, substantive advice, or only an off-topic anecdote, we tagged each comment with the types of critique that it contained. The first author tagged all of the response text, making sure to first randomize the responses across subjects and hide identifying information in the response text. In particular, with the conditions of the comments hidden, we labeled whether each comment contained a specific problem, an actionable suggestion, a compliment and/or a reaction to the video that did not fit under the other categories (e.g., "Wait, what?", "Interesting..."). Because our system supports efficiently referencing specific times and visuals in the video, we also labeled whether each comment discussed the video as a whole (i.e. global comment), or a single point in the video (i.e. local comment).

We found that all users produced more comments with our interface than when using text (Figure 6.12). On average users produced 21.9 critiques ($\sigma = 9.06$) using our interface and 10.4 ($\sigma = 4.15$) using text. A Wilcoxon Signed-rank test shows that the difference is significant ($W = 36, p < 0.05$). In addition, users of our interface produced 15.4 ($\sigma = 5.29$) comments containing specific problems compared to 7.5 ($\sigma = 4.36$) with text ($W = 28, p < 0.05$). Users also produced significantly more comments containing actionable suggestions when using our interface ($\mu = 11.9, \sigma = 4.94$) than when using text ($\mu = 5.38, \sigma = 2.64$) ($W = 28, p < 0.05$). However, users did not produce significantly more compliments or reactions. As users did not produce more global critiques, the increase in comments with our interface resulted from an increase in local critiques ($W = 36, p < 0.05$).

Overall, users reported that they found the recording interface to be more efficient and preferable to text for recording comments (Figure 6.13). In an interview, users mentioned the following advantages of the interface: overall efficiency (7 users), scrubbing instead of transcribing timestamps (6 users), providing comments without pausing the video (4 users), pointing to vi-

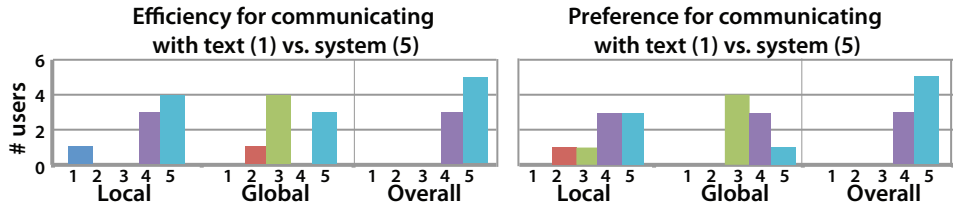

Figure 6.13: Users compare the interface and text for making a variety of comments.

sual changes instead of writing them down (4 users), referencing the source video transcript (3 users). The user who did not mention efficiency mentioned that using the interface felt more conversational than text. Though users preferred our interface for most types of critiques, our interface received the lowest ratings for delivering global feedback. This is unsurprising because the recording interface does not provide any specialized support for global comments. When asked about advantages of text, 5 users mentioned they could go back and edit their feedback (e.g. to rearrange their comments into categories), and 2 users mentioned they felt they had more time to think through what they were going to say when using text.

Viewing interface informal evaluation

To find out if video authors could use the viewing interface to quickly understand and review feedback on videos, we conducted an informal evaluation. Using school mailing lists, we recruited 4 participants (2 female, 2 male, ages 22-45) with experience receiving feedback on videos in-person and via e-mail. In particular, we recruited two professional video producers (U1,U2), and two participants currently working on video projects (U3, U4).

Method

Ahead of time, we collected videos from the two participants working on video projects, U3 and U4 (e.g. a 3 minute conference video, and a 1 minute project teaser). We also collected feedback on each project from one member of the project team using our feedback recording interface. During the informal evaluation, each participant used the feedback viewing interface to view, interpret, and judiciously filter the reviewer's suggestions into a list of changes to make on the next round of edits. U3 and U4 reviewed the feedback provided on their own video, while the two professional video editors (U1 and U2) each reviewed the feedback provided on one of these videos. We asked U1 and U2 to view the existing video ahead of time, and we told them the context of the feedback (e.g. a peer on the video team provided feedback on the editing and story of the video). After users completed the task, the users answered interview questions and Likert scale questions comparing receiving feedback with the viewing interface to prior experience receiving feedback over e-mail and in person.

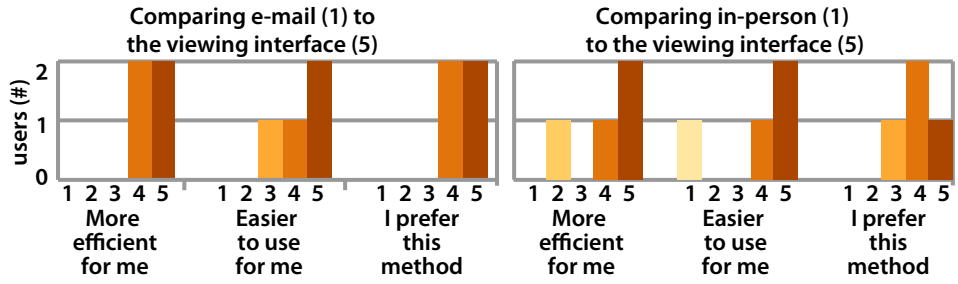

Figure 6.14: Users compared our viewing interface to existing methods of providing feedback. All users preferred our system overall when compared to e-mail.

Results: How users interacted with the feedback

All users were able to use the interface to review, interpret and filter the critiques left by the reviewer. Three users primarily navigated comments while they were sorted in the source video sort method (Table 6.3). All users read segmented comments, sometimes navigating to the corresponding frame in the source and feedback video by clicking on the comment. While most users (U1,U2,U3) only played comments that they didn't understand without context, U4 played all comments on the first pass, and read them on subsequent passes. When users didn't understand a comment after playing it, they either used the source timeline (U2, U3) or feedback session timeline (U4) to navigate to an earlier point.

Results: Interview and Likert scale questions

All users enthusiastically stated they would use the interface to receive feedback asynchronously. In a Likert scale survey, all users reported that they preferred our interface to e-mail, and that they found our interface to be more efficient than e-mail (Figure 6.14).

In the interview, all users mentioned that they preferred our interface to e-mail because comments were easier to contextualize as you could see the reviewer's playhead position while they gave the comment. Users mentioned that with e-mail, unlike our interface, they find critiques to be confusing because the context is often unclear (U1, U3, U4), and they find looking up timestamps to be tedious (U2). In addition, all users said they liked that it was easy to find more context if they needed to by navigating the source video (U1,U2,U3), viewing the source video transcript (U3), or scrubbing in the feedback timeline (U4). Users also liked being able to gauge the reviewers reaction using their tone of voice (U2, U4), the webcam video (U2), and extra reaction comments and hesitations (U2). U2, who previously taught video production, mentioned our interface was "the closest to in person feedback I've seen", and mentioned it would be great to use in online classes. U1 mentioned he would be particularly interested in using our interface instead of e-mail in cases where there are multiple similarly-skilled creative partners (e.g. a producer and an editor), or a particularly invested client (cases where everyone "deserves" to give detailed feedback).

Unsurprisingly, some users found in-person feedback to be more efficient and easier to use than our system (Figure 6.14). All users mentioned that with our interface, unlike in-person feedback, they didn't get an opportunity to immediately resolve disagreements about changes. However,

		U1	U2	U3	U4
Navigation	Feedback timeline	0	0	0	1
	Source timeline	0	1	5	0
	Shot-based timeline	0	4	3	0
	Segmented comments	2	32	84	211
Sort	Feedback session	100	0	0	0
(% of time)	Source time	0	73%	97%	98%
	Duration	0	22%	0	1%
Editing	Delete	2	8	19	39
	Edit text	0	0	0	19
	Mark complete	0	0	0	19
	Mark local/global	0	1	0	5
Search		0	0	1	0
Total time		7	12	14	19

Table 6.3: Interactions during the user study included users navigating the feedback session, sorting the feedback session comments, editing the comments, searching within the comments. Users primarily used the segmented comments for navigation, and most reviewed the comments sorted by source time (default sort is feedback session time). While all users deleted comments, only one user edited the original text of the comments.

users identified benefits of our system compared to in-person feedback including the documentation of requested changes with their timestamps (U2, U3, U4), and the capability to sort and browse feedback (U2, U3). U3 told us that having your work criticized in person can be stressful, whereas she found

We asked users about what they would like to change about the interface and all users mentioned additional tagging capabilities. Specifically, users mentioned that they would like to assign tags based on priority of tasks, mark comments that need further discussion, or type of change required to fix the problem (e.g., reshoot, audio/video quality, b-roll). U1 and U2 also mentioned that they prefer to preserve reviewers comments rather than edit them. Instead, they would like the ability to write notes along with each comment.

6.9 Limitations and future work

Our current implementation of VidCrit has some limitations.

Reviewers can't view or edit their comments: Like an in-person review, our current system does not support reviewers editing their own feedback video after it is recorded. In the recording interface study, reviewers pointed out that unlike in-person reviews, the feedback video can be replayed at a later time. Thus, reviewers wanted to remove mistakes, and ensure they do not repeat comments. With an accurate real-time transcription of the review session, reviewers could view their feedback and edit the review session via the text transcript, similar to prior

work [118, 27, 142, 143].

Receiving comments from multiple reviewers: Our system addresses one common case in which the video author receives feedback from one reviewer at a time. However, if multiple reviewers provide feedback in parallel, video authors would need to look at each reviewers feedback in a separate interface instance. In the future, we could distinguish comments from different reviewers in the viewing interface and support multiple feedback session timelines. With more than a few reviewers, we could provide an aggregate view of categorized comments, similar to prior work [84].

Improving local/global comment labelling: We label global comments using a keyword-based approach. We would prefer to use a more robust n-gram classification based approach, but global comments are rare within the dataset we collected. Out of 25-45 comments provided per video, only 2-5 of these comments addresses global issues. In the future, we will collect more data to enable better classification.

Synchronous use of our system: In this paper we focus on the problem of asynchronous review as synchronous review is not always possible. However users noted that our system provides benefits over in-person feedback in that it lets authors search, browse and skim the documented feedback. In the future we will use our system to document synchronous conversations.

6.10 Conclusion

VidCrit is a new system for commenting and receiving comments on video projects. VidCrit allows users to deliver comments asynchronously while preserving benefits of both in-person and digital text critiques. We found users provide more actionable suggestions using our recording interface. In an informal evaluation, users successfully used our viewing interface to review editor feedback and reported they preferred the interface over existing methods.

Chapter 7

Conclusion

7.1 Restatement of contributions

This thesis considers video navigation based on structured text to support searching, browsing and skimming in video. We create three systems to demonstrate the application of this video navigation technique to high level tasks. The technical and design contributions of our interactive systems can be summarized as follows:

- A system for assisting authors in creating navigable text representations of informative videos.
  - An interface for letting authors align their summaries to video using the transcript of the video.
  - A hybrid automatic and crowd approach for creating hierarchical chapter and section summaries.
  - A comparative study between our representation, navigable transcripts, and navigable timeline of the efficiency of such video summaries for a skimming task.
- A system for helping film students and professionals search, browse, and skim within film.
  - Formative interviews with film professionals exploring relevant tasks within the domain of film.
  - Automatic approaches for generating alignment between plot summaries to scripts, and captions to the base video.
  - Assessment of utility of documents for answering existing and novel queries in film studies.
- A system to facilitate the use of casually recorded feedback for communicating video revisions.
  - Formative interviews with professional and amateur video creators exploring challenges of communicating feedback.

- A feedback-providing interface and a feedback viewing interface for providing and efficiently viewing video feedback.
- A study comparing spoken and written feedback in the space of video critiques and a project-centered evaluation of our end-to-end system.

7.2 Future work

A few areas for immediate future work exist.

Analyzing large datasets using structured text

Each project focuses on searching, browsing and skimming videos using structured text segments. However, the structured text summaries for video segments could be used for other quantitative analysis or comparison tasks. For instance, we could provide tools for producing quantitative analysis for film search (similar to WordSeer in the domain of literature [96]). We could enable users to quantify current queries (e.g., how many scenes do characters appear together), and produce visual reports of this information including the related video clips. Further, we could help users compare multiple videos (e.g., adaptations of a similar film or stories with the same plot) or find patterns in movies (e.g., hero's journey). In other domains, like informational videos or how-to videos such semantic alignment may also be interesting. For video digests, we could provide links between semantically similar video clips for related lectures, or create dependency trees for completing a series of videos.

Visual features

All three projects leverage text and audio features to improve video navigation. With the exception of VidCrit automatically determining shot boundaries, none of these projects use visual features to improve video navigation or analysis. As film studies scholars and film professionals seek to understand the visual content of films, we could use visual features (e.g., shot type, camera movement, color palettes, and action/object detection) to aid search and higher-level analysis of film content. For instance, we could let users find what types of shots are used in different types of scenes (e.g., internal vs. external, chase vs. dialog). We could also use visual features to improve existing capabilities (e.g., alignment in SceneSkim, segmentation in Video Digests). For VidCrit, we could use visual features to find the differences between the current video and a video changed according to the provided comments. We could use the structured comments to search and browse a video diff.

New domains and users

We consider two existing domains and one new domain for enabling searching, browsing and skimming in videos. We could support documentation and review of other steps in the creative process (e.g., in-person feedback session), or we could align live performances (e.g., play rehearsal) to

corresponding scripts for review. In addition, such techniques could be used for recorded conversations (e.g., a doctor-patient conversation) in order to generate indexable summaries (e.g., a doctor's SOAP note). In addition, structured text may make video more accessible to users. Specifically, we could consider broadly how to tailor the text representations of conversation or video in a way that would be particularly useful to people with different needs. For instance, similar to Video Digests [106] we might author audio descriptions of inaccessible visual content in addition to summaries of the spoken content.

Generative tasks

Finally, these projects addressed using structured-text for searching, browsing, and skimming video. We could also use the structured-text alignment of video for generative tasks (e.g., write a story and illustrate it using related film clips, write a curriculum and populate it with lecture clips, use VidCrit comments to automatically edit the video).

Emerging mediums

In traditional video, we have considered how to let users search and navigate along a timeline. 360-degree video presents additional challenges as users must navigate spatially as well as temporally. We have started studying navigation in 360 video by examining where users tend to look in 360 scenes [122], and by enabling simple user interactions for locating important points [105].

Structured text may be useful for browsing 360-degree video when reviewing video footage for editing purposes, or when using 360-degree video in observational research contexts (e.g., classrooms [107]). We may be able to use automatic techniques (e.g., microphone arrays + transcriptions, recent computer vision [48]) or tagging during recording to produce text descriptions for 360-degree video.

7.3 Summary

The large problem of this thesis is that it is challenging to use video for reference and reuse. More challenging than text as the temporal element makes it opaque to users. For this problem, we propose that structured text aligned to the video can help with this problem. Such structured text representations support user's existing mental models and enable searching, browsing and skimming tasks by representing videos at a higher level conceptual structure than the traditional methods offered (e.g., frame-based and transcript-based). The three prototypes that we've explored in this thesis – Video Digests, SceneSkim and VidCrit – explore regions of this design space. Between these projects, we explore three domains: lecture videos, film and feedback, that cover three main types of videos: recorded live events, edited videos, and casually recorded spoken content. In our projects we compared our structured text representations to videos and transcripts for efficiently helping users identify main points (Chapter 4), demonstrated the ability of the systems to flexibly answer novel questions (Chapter 5), and applied video to a new domain by automatically making it navigable (Chapter 6). In the future we will apply our work to new domains, users and media,

enable new NLP tasks, and explore providing synthesis and feedback based on our higher level knowledge of video documents.

Bibliography

[1] edX. http://www.edx.org.

[2] *For a Few Days More* screenplay formatting guide. http://www.oscars.org/sites/default/files/scriptsample.pdf. Accessed 2015-03-03.

[3] *Star Wars Episodes IV – VI*: Themes, motifs and symbols. http://www.sparknotes.com/film/starwars/themes.html. Accessed 2015-04-12.

[4] Erin's famous fruit pizza. https://youtu.be/ORbZ6jzJLIA. Accessed 2016-03-09.

[5] Framebench. http://www.framebench.com/video-collaboration/. Accessed 2015-05-31.

[6] Frame.io. http://frame.io/. Accessed 2015-05-31.

[7] France vlog | annecy. http://www.leoniesii.com/annecy-france/. Accessed 2015-05-31.

[8] Khan Academy. http://khanacademy.org.

[9] Learnkendama.com kickstarter video. https://youtu.be/MkkuCJ1l2K8. Accessed 2016-03-09.

[10] Marqueed. https://www.marqueed.com/. Accessed 2016-03-09.

[11] rev.com. https://www.rev.com/. Accessed 2016-03-21.

[12] Screenlight. https://screenlight.tv/features/. Accessed 2015-05-31.

[13] Silverback. https://silverbackapp.com/. Accessed 2016-03-21.

[14] Skintillates video. https://vimeo.com/165809373. Accessed 2016-07-20.

[15] Ted. http://www.ted.com/. Accessed 2015-04-11.

[16] Usertesting. https://www.usertesting.com/product/videos-and-metrics. Accessed 2016-03-21.

[17] Wipster. http://wipster.io/. Accessed 2015-05-31.

[18] Abowd, G. D., Gauger, M., and Lachenmann, A. The family video archive: an annotation and browsing environment for home movies. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, ACM (2003), 1–8.

[19] Apple tv. https://www.apple.com/tv/, Jan. 2019.

[20] Baio, A., and Bell-Smith, M. supercut.org. http://supercut.org/. Accessed 2015-07-17.

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

[22] Barnes, C., Goldman, D. B., Shechtman, E., and Finkelstein, A. Video tapestries with continuous temporal zoom. *ACM Trans. Graph. 29*, 4 (July 2010), 89:1–89:9.

[23] Bartindale, T., Schofield, G., and Wright, P. C. Tryfilm: Situated support for interactive media productions. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, ACM (2016), 1410–1420.

[24] Bartindale, T., Sheikh, A., Taylor, N., Wright, P., and Olivier, P. Storycrate: tabletop storyboarding for live film production. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2012), 169–178.

[25] Bernstein, M. S., Brandt, J., Miller, R. C., and Karger, D. R. Crowds in two seconds: Enabling realtime crowd-powered interfaces. In *UIST*, ACM (2011), 33–42.

[26] Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., and Panovich, K. Soylent: a word processor with a crowd inside. In *Proc. of the 23nd annual*, ACM (2010), 313–322.

[27] Berthouzoz, F., Li, W., and Agrawala, M. Tools for placing cuts and transitions in interview video. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 67.

[28] Berthouzoz, F., Li, W., and Agrawala, M. Tools for placing cuts and transitions in interview video. *ACM Trans.Graph. 31*, 4 (2012), 67.

[29] Bird, S. NLTK: The natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, Association for Computational Linguistics (2006), 69–72.

[30] Bojanowski, P., Bach, F., Laptev, I., Ponce, J., Schmid, C., and Sivic, J. Finding actors and actions in movies. In *Proc. IEEE International Conference on Computer Vision* (2013).

[31] Boreczky, J., Girgensohn, A., Golovchinsky, G., and Uchihashi, S. An interactive comic book presentation for exploring video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, ACM (New York, NY, USA, 2000), 185–192.

[32] Brode, D., and Deyneka, L. *Sex, Politics, and Religion in Star Wars: An Anthology*. Scarecrow Press, 2012.

[33] Burrows, S., Potthast, M., and Stein, B. Paraphrase acquisition via crowdsourcing and machine learning. *ACM Transactions on Intelligent Systems and Technology (TIST) 4*, 3 (2013), 43.

[34] Buzek, O., Resnik, P., and Bederson, B. B. Error driven paraphrase annotation using mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, Association for Computational Linguistics (2010), 217–221.

[35] Casares, J., Long, A. C., Myers, B. A., Bhatnagar, R., Stevens, S. M., Dabbish, L., Yocum, D., and Corbett, A. Simplifying video editing using metadata. In *Proceedings of DIS*, ACM (2002), 157–166.

[36] Cattelan, R. G., Teixeira, C., Goularte, R., and Pimentel, M. D. G. C. Watch-and-comment as a paradigm toward ubiquitous interactive video editing. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 4*, 4 (2008), 28.

[37] Chi, P.-Y., Ahn, S., Ren, A., Dontcheva, M., Li, W., and Hartmann, B. MixT: Automatic generation of step-by-step mixed media tutorials. In *Proceedings of UIST*, ACM (2012), 93–102.

[38] Chi, P.-Y., Liu, J., Linder, J., Dontcheva, M., Li, W., and Hartmann, B. Democut: generating concise instructional videos for physical demonstrations. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ACM (2013), 141–150.

[39] Choi, F. Y. Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, Association for Computational Linguistics (2000), 26–33.

[40] Christel, M. G., Smith, M. A., Taylor, C. R., and Winkler, D. B. Evolving video skims into useful multimedia abstractions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press/Addison-Wesley Publishing Co. (1998), 171–178.

[41] Cisco, V. Cisco visual networking index: Forecast and trends, 2017–2022. *White Paper* (2018).

[42] Cockburn, A., and Dale, T. Ceva: a tool for collaborative video analysis. In *Proceedings of the international ACM SIGGROUP conference on Supporting group work: the integration challenge*, ACM (1997), 47–55.

[43] Cohen, J., Withgott, M., and Piernot, P. Logjam: a tangible multi-person interface for video logging. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, ACM (1999), 128–135.

[44] Cour, T., Jordan, C., Miltsakaki, E., and Taskar, B. Movie/script: Alignment and parsing of video and text transcription. In *Computer Vision–ECCV 2008*. Springer, 2008, 158–171.

[45] Decker, K., and Eberl, J. *Star Wars and Philosophy: More Powerful Than You Can Possibly Imagine*. Popular culture and philosophy. Open Court, 2005.

[46] Denkowski, M., Al-Haj, H., and Lavie, A. Turker-assisted paraphrasing for english-arabic machine translation. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, Association for Computational Linguistics (2010), 66–70.

[47] Diakopoulos, N., and Essa, I. Videotater: an approach for pen-based digital video segmentation and tagging. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, ACM (2006), 221–224.

[48] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), 2625–2634.

[49] Du, L., Buntine, W., and Johnson, M. Topic segmentation with a structured topic model. In *Proceedings of NAACL-HLT* (2013), 190–200.

[50] Eisenstein, J., and Barzilay, R. Bayesian unsupervised topic segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics (2008), 334–343.

[51] Everingham, M., Sivic, J., and Zisserman, A. Hello! My name is... Buffy – automatic naming of characters in TV video. In *British Machine Vision Conference* (2006).

[52] Facebook watch. https://www.facebook.com/watch/, Jan. 2019.

[53] Final cut pro. https://www.adobe.com/products/premiere.html, Jan. 2019.

[54] Fouse, A., Weibel, N., Hutchins, E., and Hollan, J. D. Chronoviz: A system for supporting navigation of time-coded data. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, ACM (2011), 299–304.

[55] Fritz, B. OMDB API. http://www.omdbapi.com/. Accessed 2015-03-03.

[56] Gargi, U., Kasturi, R., and Strayer, S. H. Performance characterization of video-shot-change detection methods. *Circuits and Systems for Video Technology, IEEE Transactions on 10*, 1 (2000), 1–13.

[57] Gendler, T. Philosophy 181: Introduction. http://oyc.yale.edu/philosophy/phil-181/lecture-1, Spring 2011.

[58] Goldman, D. B., Curless, B., Seitz, S. M., and Salesin, D. Schematic storyboarding for video visualization and editing. *ACM Transactions on Graphics (Proc. SIGGRAPH) 25*, 3 (2006).

[59] Goldman, S. R., and Rakestraw, J. A. Structural aspects of constructing meaning from text. *Handbook of reading research 3*, 1 (2000), 311–335.

[60] Google cloud speech to text. https://cloud.google.com/speech-to-text/, May 2019.

[61] Grossman, T., Matejka, J., and Fitzmaurice, G. Chronicle: Capture, exploration, and playback of document workflow histories. In *Proceedings of UIST*, ACM (2010), 143–152.

[62] Guo, P. J., Kim, J., and Rubin, R. How video production affects student engagement: An empirical study of mooc videos. In *Proceedings of the first ACM Learning@ scale conference*, ACM (2014), 41–50.

[63] Gupta, V., and Lehal, G. S. A survey of text summarization extractive techniques. *Journal of Emerging Technologies in Web Intelligence 2*, 3 (2010), 258–268.

[64] Hartley, J., and Trueman, M. A research strategy for text desigbers: The role of headings. *Instructional Science 14*, 2 (1985), 99–155.

[65] Haubold, A., and Kender, J. R. Augmented segmentation and visualization for presentation videos. In *Proceedings of the 13th annual ACM international conference on Multimedia*, ACM (2005), 51–60.

[66] Haubold, A., and Kender, J. R. VAST MM: Multimedia browser for presentation video. In *Proceedings of CIVR*, ACM (2007), 41–48.

[67] He, L., Sanocki, E., Gupta, A., and Grudin, J. Auto-summarization of audio-video presentations. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, ACM (1999), 489–498.

[68] Hearst, M. A. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational linguistics 23*, 1 (1997), 33–64.

[69] Jackson, D., Nicholson, J., Stoeckigt, G., Wrobel, R., Thieme, A., and Olivier, P. Panopticon: A parallel video overview system. In *Proceedings of UIST*, ACM (2013), 123–130.

[70] Jones, D. A., Wolf, F., Gibson, E., Williams, E., Fedorenko, E., Reynolds, D. A., and Zissman, M. Measuring the readability of automatic speech-to-text transcripts. In *Eighth European Conference on Speech Communication and Technology* (2003).

[71] Khan, S. Us history overview: Jamestown to the civil war. https://www.khanacademy.org/, April 2011.

[72] Kim, J., Guo, P. J., Cai, C. J., Li, S.-W. D., Gajos, K. Z., and Miller, R. C. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of UIST*, ACM (2014), 563–572.

[73] Kim, J., Nguyen, P., Weir, S., Guo, P. J., Miller, R. C., and Gajos, K. Z. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the 2014 ACM annual conference on Human factors in computing systems*, ACM (2014).

[74] Kim, J., Nguyen, P. T., Weir, S., Guo, P. J., Miller, R. C., and Gajos, K. Z. Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of CHI*, ACM (2014), 4017–4026.

[75] Kim, J., Shang-Wen, L. D., Cai, C. J., Gajos, K. Z., and Miller, R. C. Leveraging video interaction data and content analysis to improve video learning. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*, ACM (2014).

[76] Kirk, D., Sellen, A., Harper, R., and Wood, K. Understanding videowork. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2007), 61–70.

[77] Klemmer, S. The power of prototyping. https://class.coursera.org/hci/lecture, 2012.

[78] Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, IEEE (2008), 1–8.

[79] Lasecki, W., Miller, C., Sadilek, A., Abumoussa, A., Borrello, D., Kushalnagar, R., and Bigham, J. Real-time captioning by groups of non-experts. In *UIST*, ACM (2012), 23–34.

[80] Lasecki, W. S., Song, Y. C., Kautz, H., and Bigham, J. P. Real-time crowd labeling for deployable activity recognition. In *Proceedings of the 2013 conference on Computer supported cooperative work*, ACM (2013), 1203–1212.

[81] Lavigne, S. Videogrep. http://lav.io/2014/06/videogrep-automatic-supercuts-with-python/. Accessed 2015-05-31.

[82] Leake, M., Davis, A., Truong, A., and Agrawala, M. Computational video editing for dialogue-driven scenes. *ACM Trans. Graph. 36*, 4 (2017), 130–1.

[83] Lorch, R. F. Text-signaling devices and their effects on reading and memory processes. *Educational psychology review 1*, 3 (1989), 209–234.

[84] Luther, K., Tolentino, J.-L., Wu, W., Pavel, A., Bailey, B. P., Agrawala, M., Hartmann, B., and Dow, S. P. Structuring, aggregating, and evaluating crowdsourced design critique. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, ACM (2015), 473–485.

[85] Mackay, W. E. Eva: An experimental video annotator for symbolic analysis of video data. *Acm Sigchi Bulletin 21*, 2 (1989), 68–71.

[86] Mackay, W. E., and Beaudouin-Lafon, M. Diva: Exploratory data analysis with multimedia streams. In *Proceedings of the SIGCHI conference on human factors in computing systems*, ACM Press/Addison-Wesley Publishing Co. (1998), 416–423.

[87] Malioutov, I., and Barzilay, R. Minimum cut model for spoken lecture segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, Association for Computational Linguistics (2006), 25–32.

[88] Mas, J., and Fernandez, G. Video shot boundary detection based on color histogram. *Notebook Papers TRECVID2003, Gaithersburg, Maryland, NIST* (2003).

[89] Matejka, J., Grossman, T., and Fitzmaurice, G. Ambient help. In *Proceedings of CHI*, ACM (2011), 2751–2760.

[90] Matejka, J., Grossman, T., and Fitzmaurice, G. Swift: Reducing the effects of latency in online video scrubbing. In *Proceedings of CHI*, ACM (2012), 637–646.

[91] Matejka, J., Grossman, T., and Fitzmaurice, G. Swifter: Improved online video scrubbing. In *Proceedings of CHI*, ACM (2013), 1159–1168.

[92] Matejka, J., Grossman, T., and Fitzmaurice, G. Video lens: Rapid playback and exploration of large video collections and associated metadata. In *Proceedings of UIST*, ACM (2014), 541–550.

[93] Mayer, R. E., and Moreno, R. Nine ways to reduce cognitive load in multimedia learning. *Educational psychologist 38*, 1 (2003), 43–52.

[94] Mohamad Ali, N., Smeaton, A. F., and Lee, H. Designing an interface for a digital movie browsing system in the film studies domain. *International Journal of Digital Content Technology and Its Applications 5*, 9 (2011), 361–370.

[95] Monserrat, T.-J. K. P., Zhao, S., McGee, K., and Pandey, A. V. NoteVideo: Facilitating navigation of blackboard-style lecture videos. In *Proceedings of CHI*, ACM (2013), 1139–1148.

[96] Muralidharan, A., Hearst, M. A., and Fan, C. Wordseer: a knowledge synthesis environment for textual data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, ACM (2013), 2533–2536.

[97] Needleman, S. B., and Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology 48*, 3 (1970), 443–453.

[98] Nenkova, A., Maskey, S., and Liu, Y. Automatic summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts of ACL 2011*, Association for Computational Linguistics (2011), 3.

[99] Noah, D. B. B. O., and Smith, A. Learning latent personas of film characters. ACL (2013).

[100] Norman, D. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[101] Ochshorn, R. M., and Hawkins, M. Gentle.

[102] Olsen, D. R., Partridge, B., and Lynn, S. Time warp sports for internet television. *ACM Transactions on Computer-Human Interaction (TOCHI) 17*, 4 (2010), 16.

[103] Pavel, A., Goldman, D. B., Hartmann, B., and Agrawala, M. Sceneskim: Searching and browsing movies using synchronized captions, scripts and plot summaries. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, ACM (2015), 181–190.

[104] Pavel, A., Goldman, D. B., Hartmann, B., and Agrawala, M. Vidcrit: Video-based asynchronous video review. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, ACM (2016), 517–528.

[105] Pavel, A., Hartmann, B., and Agrawala, M. Shot orientation controls for interactive cinematography with 360 video. In *In submission to UIST 2017*,

[106] Pavel, A., Reed, C., Hartmann, B., and Agrawala, M. Video digests: a browsable, skimmable format for informational lecture videos. In *UIST*, ACM (2014), 573–582.

[107] Pea, R., Mills, M., Rosen, J., Dauber, K., Effelsberg, W., and Hoffert, E. The diver project: Interactive digital video repurposing. *IEEE multimedia 11*, 1 (2004), 54–61.

[108] Pearson, P. D., Kamil, M. L., Mosenthal, P. B., Barr, R., et al. *Handbook of reading research*. Routledge, 2016.

[109] Pongnumkul, S., Dontcheva, M., Li, W., Wang, J., Bourdev, L., Avidan, S., and Cohen, M. F. Pause-and-play: Automatically linking screencast video tutorials with applications. In *Proceedings of UIST*, ACM (2011), 135–144.

[110] Adobe premeire. https://www.adobe.com/products/premiere.html, Jan. 2019.

[111] Quicktime video player. https://support.apple.com/quicktime, Jan. 2019.

[112] Ramanathan, V., Joulin, A., Liang, P., and Fei-Fei, L. Linking people in videos with "their" names using coreference resolution. In *Computer Vision–ECCV 2014*. Springer, 2014, 95–110.

[113] Ramos, G., and Balakrishnan, R. Fluid interaction techniques for the control and annotation of digital video. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, ACM (2003), 105–114.

[114] Roku tv. https://www.roku.com/products/roku-tv, Jan. 2019.

[115] Ronfard, R. Reading movies: An integrated DVD player for browsing movies and their scripts. In *Proceedings of the 12th annual ACM international conference on Multimedia*, ACM (2004), 740–741.

[116] Ronfard, R., and Thuong, T. T. A framework for aligning and indexing movies with their script. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, vol. 1, IEEE (2003), I–21.

[117] Rosling, H. The best statistics you've ever seen. http://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen, February 2006.

[118] Rubin, S., Berthouzoz, F., Mysore, G. J., Li, W., and Agrawala, M. Content-based tools for editing audio stories. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ACM (2013), 113–122.

[119] Rubin, S., Berthouzoz, F., Mysore, G. J., Li, W., and Agrawala, M. Content-based tools for editing audio stories. In *Proceedings of UIST*, ACM (2013), 113–122.

[120] Schoeffmann, K., Hudelist, M. A., and Huber, J. Video interaction tools: a survey of recent work. *ACM Computing Surveys (CSUR) 48*, 1 (2015), 14.

[121] Silvio, C., Vinci, T., Palumbo, D., and Sullivan, C. *Culture, Identities and Technology in the Star Wars Films: Essays on the Two Trilogies*. Critical Explorations in Science Fiction and Fantasy. McFarland & Company, 2007.

[122] Sitzmann, V., Serrano, A., Pavel, A., Agrawala, M., Gutierrez, D., and Wetzstein, G. Saliency in vr: How do people explore virtual environments? In *In submission to SIGGRAPH ASIA 2017*,

[123] Smeaton, A. F., Over, P., and Doherty, A. R. Video shot boundary detection: Seven years of trecvid activity. *Computer Vision and Image Understanding 114*, 4 (2010), 411–418.

[124] Smith, M. A., and Kanade, T. Video skimming and characterization through the combination of image and language understanding. In *Content-Based Access of Image and Video Database, 1998. Proceedings., 1998 IEEE International Workshop on*, IEEE (1998), 61–70.

[125] Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation 28*, 1 (1972), 11–21.

[126] Stark, H. A. What do paragraph markings do? *Discourse processes 11*, 3 (1988), 275–303.

[127] Tang, A., and Boring, S. # epicplay: crowd-sourcing sports video highlights. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2012), 1569–1572.

[128] Tapaswi, M., Bäuml, M., and Stiefelhagen, R. Aligning plot synopses to videos for story-based retrieval. *International Journal of Multimedia Information Retrieval 4*, 1 (2015), 3–16.

[129] Tapaswi, M., Bäuml, M., and Stiefelhagen, R. Book2movie: Aligning video scenes with book chapters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), 1827–1835.

[130] Taskiran, C. M., Pizlo, Z., Amir, A., Ponceleon, D., and Delp, E. J. Automated video program summarization using speech transcripts. *Multimedia, IEEE Transactions on 8*, 4 (2006), 775–791.

[131] Tiedemann, J. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, European Language Resources Association (ELRA) (2012).

[132] Truong, A., Berthouzoz, F., Li, W., and Agrawala, M. Quickcut: An interactive tool for editing narrated video. In *Proc. UIST'16*, ACM (2016), To Appear.

[133] Truong, A., Berthouzoz, F., Li, W., and Agrawala, M. Quickcut: An interactive tool for editing narrated video. In *Proc. UIST*, vol. 16 (2016).

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

[135] Uchihashi, S., Foote, J., Girgensohn, A., and Boreczky, J. Video manga: Generating semantically meaningful video summaries. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, ACM (1999), 383–392.

[136] Victor, B. Media for thinking the unthinkable. http://worrydream.com/MediaForThinkingTheUnthinkable, April 2013.

[137] Victor, B. Personal communication, December 2013.

[138] Vimeo video player. http://vimeo.com, Apr. 2015.

[139] Volkmer, T., Smith, J. R., and Natsev, A. P. A web-based system for collaborative annotation of large image and video collections: an evaluation and user study. In *Proceedings of the 13th annual ACM international conference on Multimedia*, ACM (2005), 892–901.

[140] Weher, K., and Poon, A. Marquee: A tool for real-time video logging. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (1994), 58–64.

[141] Whittaker, S., and Amento, B. Semantic speech editing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM (2004), 527–534.

[142] Yoon, D., Chen, N., Guimbretière, F., and Sellen, A. Richreview: blending ink, speech, and gesture to support collaborative document review. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, ACM (2014), 481–490.

[143] Yoon, D., Chen, N., Randles, B., Cheatle, A., Löckenhoff, C. E., Jackson, S. J., Sellen, A., and Guimbretière, F. Richreview++: Deployment of a collaborative multi-modal annotation system for instructor feedback and peer discussion. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, ACM (2016), 195–205.

[144] Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Valtchev, V., and Woodland, P. *The HTK Book*. Cambridge University Engineering Department, 2002.

[145] Youtube video player. http://www.youtube.com, Jan. 2019.

[146] Yuan, J., and Liberman, M. Speaker identification on the SCOTUS corpus. *Journal of the Acoustical Society of America 123*, 5 (2008), 3878.

[147] Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *arXiv preprint arXiv:1506.06724* (2015).