Indexing and Browsing Unstructured Videos using Visual, Audio, Textual, and Facial Cues

Alexander Haubold

Submitted in partial fulfillment of the Requirements for the degree of Doctor of Philosophy in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY
2008
Abstract

Indexing and Browsing Unstructured Videos using Visual, Audio, Textual, and Facial Cues

Alexander Haubold

In the domain of highly unstructured and unedited videos, we introduce novel approaches for automatic indexing and browsing of audio, visual, and textual contents. Our video browser is one of the first of its kind for large libraries of unstructured video data, having been tested on more than 500 university lecture and student presentation videos. Unique multi-modal indices sensitive to the raw video data and to user interaction have been measured and refined in extensive user studies with more than 1,000 students. We report on significant improvements in typical video retrieval tasks, such as search by text and visual contents. We also report on statistically significant improvements in student performance when the video browser was used for exam preparation.

We introduce the prototype platform VAST MM (Video Audio Structure Text MultiMedia), which encompasses a video indexer and a platform independent video browser, featuring visual cues, a custom-designed MPEG1 video stream player, text search, and comparison tools based on key phrases. Relevant indices shared among video genres include visual-based segmentation of likely scenes; automatic speech recognition (ASR) transcripts; and filters for ASR text derived from external material. Indices applicable to student presentation videos include audio-based speaker segmentation; a graphical face index of speakers in the video; and audio-based speaker clustering of
recurring speakers. Indices unique to lecture videos include visual-based clustering of scenes into teaching units; and visual-based classification of video keyframes.

The VAST MM browser demonstrates how multi-modal information can be combined to provide a rich set of indices that improve the video browsing experience for users, even when the video is unedited. The timeline ties together these modalities to maintain context, while individual visual indices can be manipulated interactively by the user to adjust the amount of displayed information. In lieu of a tag cloud, we show ranked words and phrases tied to their temporal occurrence, while clustering their recurrences visually. The streaming video player is augmented with “keyframes,” redefined here as visually distinct images representative of the video’s content. We introduce face indices to replace otherwise expensive and inaccurate person recognition. Finally, with a database of more than 500 hours of video, VAST MM demonstrates a tested application with large amounts of data.

Observations have been collected from more than 1,000 students in the course of their studies to measure usefulness of video indices, ease of use of browsing tools, and impact of the availability of such resources. We find, among other results, that the presence of streaming video negatively impacts content search performance by requiring significantly more time (up to 74%) for task completion without providing higher successful task completion rates (which remain at 90%). In general, in the course of 11 experiments in a three-year period over which index cues and user interfaces were refined, we were able to significantly improve search performance for difficult tasks by decreasing required time from 436 seconds to 128 seconds, while increasing completion rates from 57% to 97%.
# Table of Contents

List of Figures

v

Acknowledgements

viii

Chapter 1  Introduction  1

Chapter 2  Prior Work  13

2.1 Segmentation of Video ................................................................. 13

2.2 Segmentation of Audio ................................................................. 14

2.3 Segmentation of Text ................................................................. 15

2.4 Segmentation of Events ............................................................. 15

2.5 User Interfaces ........................................................................... 16

2.6 Capture of Structure ................................................................. 19

Chapter 3  VAST MM: An Orientation  21

3.1 Process ..................................................................................... 22

3.2 Why it is Necessary ................................................................. 23

3.3 Evolution in Pictures ............................................................... 25
Chapter 5  Presentation Videos

5.1 Visual Indices ...................................................................................................... 96
  5.1.1 Visual Feature Extraction .......................................................................... 96
  5.1.2 Visual Feature Control ............................................................................... 100
5.2 Audio Indices ...................................................................................................... 102
  5.2.1 Speaker Segmentation ................................................................................ 102
  5.2.2 Speaker Clustering ..................................................................................... 106
    5.2.2.1 KL2 Clustering Approach .............................................................. 107
    5.2.2.2 KL2 Simulation .............................................................................. 107
    5.2.2.3 Observations .................................................................................. 108
    5.2.2.4 Empirical Solution ......................................................................... 110
    5.2.2.5 Results ............................................................................................ 111
    5.2.2.6 Speaker Clustering using VQ is not as Effective ....................... 113
  5.2.3 Visual Face Index ...................................................................................... 114
    5.2.3.1 Selecting Visual Cues .................................................................... 115
5.3 Textual Indices .................................................................................................... 119
  5.3.1 Textual Feature Extraction ......................................................................... 119
  5.3.2 Textual Feature Display ............................................................................. 122
5.4 Task-Based Experiments and Evaluation ........................................................... 123
  5.4.1 Structure in Video ...................................................................................... 125
  5.4.2 Availability of Actual Video is Counterproductive ................................... 128
  5.4.3 Face Indices Save Time ........................................................................... 130
  5.4.4 In-class Use is More Efficient .................................................................. 131
  5.4.5 VAST MM Improves Over Time ............................................................ 133
  5.4.6 Users Improve Over Time ........................................................................ 133
  5.4.7 Cues improve Time of Task Completion ................................................ 134
5.5 Conclusions.......................................................................................................... 135
List of Figures

2.1 Semantic Exploration of Lectures ................................................................. 18
2.2 Classroom 2000 ......................................................................................... 18
2.3 Video Archive Access .................................................................................. 18
2.4 Keyframe-Based UI for Digital Video ......................................................... 18
2.5 SCAN: Retrieval from Speech Archives ....................................................... 19
2.6 Video Mail Retrieval ................................................................................... 19
3.1 Flowchart of Presentation Video Analysis .................................................. 23
3.2 VAST MM Browser Version 1 ................................................................. 26
3.3 VAST MM Browser Version 2 ................................................................. 27
3.4 VAST MM Browser Version 3 ................................................................. 29
3.5 VAST MM Features over Time ................................................................. 30
4.1 Lecture Video Browser: Topological Index with Keyframe Summary .......... 34
4.2 Lecture Video Browser ............................................................................. 35
4.3 Visual Feature Filters for Classification by Media Type ............................ 37
4.4 Six Media Types ....................................................................................... 39
4.5 Decision Graph for Re-classifying End Keyframes .................................... 41
4.6 Topological Clustering ............................................................................ 43
4.7 Filters for Media Type "Board" ................................................................. 44
4.8 Filters for Media Type "Sheet" ................................................................. 45
4.9 Content Pixels for Frame Matching ......................................................... 47
4.10 Frame-Pair Matching Algorithm ................................................................. 48
4.11 Matching Heuristics ..................................................................................... 50
4.12 Development of Contents on Media Types "Sheet" and "Board" ..................... 50
4.13 Matching with near-Linear Cost ................................................................. 52
4.14 Index Phrase Repetitions in Lecture Transcripts and Textbooks ............... 61
4.15 Transcript Index Map (1) ............................................................................. 71
4.16 Transcript Index Map (2) ............................................................................. 71
4.17 Chapter Transcript Match (1) ...................................................................... 73
4.18 Chapter Transcript Match (2) ...................................................................... 73
4.19 Chapter Transcript Matching ....................................................................... 75
4.20 Multidimensional Scaling of Transcript Similarity ....................................... 76
4.21 Early Lecture Video Browser used for User Study ...................................... 80
4.22 Diagram for Intra- and Inter-Transitions in Lecture Video Browser .......... 84
4.23 New Lecture Video Browser Interface ....................................................... 85
4.24 VAST MM Video Browser used during Final Exam Preparation ............... 88
4.25 Midterm-to-Final Exam Grade Improvements ........................................... 90
4.26 Exam Grade Improvement ......................................................................... 91
5.1 Examples of Imagery from Typical Presentation Video ................................ 94
5.2 Split Panoramic View of Presentation Classroom ........................................ 94
5.3 Mid-Level Abrupt Visual Change Detection ................................................ 97
5.4 Low-Level Gradual Visual Change Measure .............................................. 98
5.5 Mid-Level Gradual Visual Change Detection .............................................. 99
5.6 Video Summary with Parameter "Scene Segmentation" (1) ......................... 101
5.7 Video Summary with Parameter "Scene Segmentation" (2) ......................... 101
5.8 Video Summary with Parameter "Zoom" (1) .................................................. 101
5.9 Video Summary with Parameter "Zoom" (2) .................................................. 101
5.10 Speaker Segmentation via BIC ................................................................. 104
5.11 Speaker Change Detection via BIC ............................................................ 104
5.12 KL2 Distance Length Effect ....................................................................... 109
5.13 Real Speech Data ...................................................................................... 109
Acknowledgements

Firstly, I would like to thank my dissertation committee – John Kender, Julia Hirschberg, Steven Feiner, Shih-Fu Chang, and Apostol (Paul) Natsev – for their time and for their interesting suggestions and comments. In particular, I would like to thank my advisor, John Kender, for his many years of support, supervision, and flexibility as I crafted my doctoral work. I am grateful for our many fruitful discussions and his skilled guidance in teaching me how to effectively conduct research, evaluate results, and present the work scientifically.

Over many years of research, studying, and teaching at Columbia University, I would like to thank an army of people, which have been supportive and helpful in many different ways:

- The faculty and staff of the School of Engineering and Applied Science Office of the Dean, who have strongly supported my studies and my research. I am grateful for the trust they have placed in me, allowing me to work freely and unrestrictedly. Their support over many years has been paramount in advancing my research and evaluating it in real-world scenarios.
• The more than 1,000 students in the required first-year engineering course, who have evaluated my work with critical yet insightful eyes over a period of 3 years. I am equally thankful for the opportunity to teach introductory Matlab to these same students – their critical thoughts on my instructional style consequently helped me to gain better insight into teaching.

• Bob Stark and Walter Khan from the Department of Mechanical Engineering Machine Shop, who have supported my crazy hobby of building the first-of-a-kind LED Lamp between 2001 and 2003, and who have frequently thereafter been a source of down-to-earth chats. Thanks for keeping things real! In connection to the LED Lamp, I would also like to thank John Kazana, who has allowed me to use facilities in the Department of Electrical Engineering for many sleepless nights of soldering.

• Various faculty members in the Department of Computer Science, who have been supportive of my user studies conducted in their courses, including Luis Gravano, Tony Jebara, Luca Carloni, and Stephen Edwards.

• Grace Chung and her staff at the Columbia Video Network (CVN) for being supportive of my research on lecture videos, providing data, and allowing me to conduct my user evaluation without any restrictions or hassle.

• The staff of the Visual Media Center, previously the Media Center for Art History, whom I worked with for 10 years on various information storage and retrieval projects. Their views and colorful conversations have always been a source of new perspectives, insight, and entertainment.
I would also like to thank members of the IBM Watson Research Center in Hawthorne – Apostol (Paul) Natsev, Milind Naphade, Murray Campbell, and John Smith – whom I have worked with for 3 years (including two summer internships). Their work and perspectives have opened a horizon of possibilities for me, providing me with real-world experience and insight into the research community.

Finally, I would like to extend my deepest thanks to close friends and family, who shape the world around me every day. Foremost, thanks to my parents, Barbara and Hans, who have given me unconditional love and support! Regardless of how grim a situation might have appeared, they have always readily extended guiding wisdom and flexibility. My heartfelt gratitude extends to my grandparents, who have been a source of joyful and carefree childhood memories.

Lastly, I would to thank my fiancé, Promiti, for her unconditional love, her support throughout my doctoral work, and her tolerance for my seemingly crazy hobbies, most of which she participates in out of curiosity of what makes me tick.
Too VAST
Chapter 1

Introduction

Videos are rich in multimedia content and semantics, which should be used by video browsers to better present the audio-visual information to the viewer. Through analysis of audio and video tracks, it is possible to extract text transcripts from audio, displayed text from video, and higher-level semantics through speaker identification and scene analysis. External data sources, when available, can be used to cross-reference the video content and impose a structure for organization. Various research tools have addressed video summarization and browsing using one or more of these modalities; however, most of them assume edited videos as input. Moreover, presently available video players and video library browsers implement few if any of the proven methods for improved video browsing.

In this thesis, we address analysis, index and search approaches, and UI design for a video browser for lecture, student presentation, and student meeting/interaction videos.
Videos of these three genres are generally unedited, are not professionally captured, and contain slowly evolving material. Indexing of such information is crucial for search and retrieval, in particular because these video archives grow rapidly and users are generally interested in very specific contents. For example, one semester of a university course is captured in roughly 40 hours of video, which is an unwieldy amount of data to search linearly for a short discussion. One semester of student presentations in a team-oriented course with \( \approx 150 \) students can easily exceed 15 hours of video-recorded material, which is equally as difficult to search for an individual student’s performance. Manual indexing of this information, however, is prohibitively expensive and cannot be justified due to the small audience it serves. Automatic indexing and browsing tools are therefore crucial in making available these audio-visual resources.

Lecture videos contain recorded material from university courses conducted primarily for on-campus students in classrooms, which are modified to feature high quality cameras and other capture equipment. A semi-professional operator, usually a student trained by the permanent staff, controls the cameras remotely, while observing the classroom plot from a control room. Instructors occasionally dictate which camera should be used and what content it should capture. Throughout the recording, a wearable wireless microphone captures the instructor’s speech. While lecture videos do not undergo post-editing, the camera operator introduces visual cuts by switching between cameras during recording. The extent of capture equipment beyond cameras is mainly driven by instructional style. Across a variety of computer science and electrical engineering courses, we have observed that pre-2004, the most common presentation
tools included the blackboard and hand-drawn sheets of paper, both of which were captured by cameras. Digital media, such as electronic presentation slides and web pages were used sparingly, and usually only to illuminate examples. More recent classroom lectures, whether or not they are recorded, exhibit the opposite trend – electronic slides and whiteboards are the dominant means of visual presentation. These observations do not, however, generalize across all fields of engineering and certainly not across different schools. Our early research on lecture material organization based on hand-produced visual cues therefore remains generally relevant.

Student presentation videos feature project presentations held by student teams as part of a course. Student presentations have gained popularity only in recent years as university instruction shifted towards team-oriented project work and presentation performance became a measured and trained professional skill. Unlike lecture videos, student presentation videos are recorded in a conventional classroom without recording-optimized conditions or special technological considerations beyond the availability of a projector. Student teams typically make use of electronic slides while presenting to the class, instructors, and visitors. A non-professional videographer, usually a student herself, operates a semi-professional camera, making all cinematographic decisions while recording. The resulting videos feature a myriad of contents, some focusing entirely on the presenter, others shifting frequently between close-ups of projected slides and far shots of the classroom stage.

Finally, student meeting/interaction videos encompass a variety of student-produced material, capturing a student team’s process throughout their semester project.
These videos are intended for introspection by the team to improve individual interaction, and to provide interesting audio-visual material for their final presentation. Course guidelines suggest recorded material such as team meetings, interviews with project clients, and field trips; however, actual content selection is completely at the discretion of the students. Students use low-end digital camcorders and selectively external microphones. Naturally, the video footage spans a barrage of audio-visual qualities, from mostly constant visual information during round-table meetings when the camera was secured on a tripod, to wildly moving imagery when students engaged in outdoor recording with hand-held equipment. The quality of speech recording for these videos is generally too low for automatic speech transcription, although good enough for human recognition. With their audio-visual characteristics, student meeting/interaction videos are most similar to home videos.

Conventional methods of content retrieval through directory-like lists and standard media players fail for large collections of videos for these three genres. Without manually or automatically generated indices, the amount of time required to find content is linearly proportional to the number of videos. Even when a candidate subset of videos has been identified, standard media players offer no functionality beyond play/pause/stop and location sliders, leaving the user with a linear browsing task to find content within a video. In our work we address the automatic creation of searchable index cues for large collections of videos and approaches of presenting multi-modal video content in a novel and interactive video browser. We specifically address the problem of generating high-quality searchable text indices from highly imperfect automatically generated transcripts.
We also address various approaches for extracting structural cues from videos, which can be used to improve the presentation of audio, visual, and textual information. One of the genres considered (lecture video) has been explored in many prior works, while an additional genre (presentation video) is newly introduced. The findings and implementation of this work find widespread application, in particular in environments where vigorous personal interaction plays an important role, for example in the team-oriented university classroom.

This work contributes new methods and approaches of multi-modal video analysis for personal interaction videos. They include:

- In the domain of lecture videos:
  - Classification and clustering of video scenes, and user interfaces for browsing this content.
  - Methods of filtering imperfect transcripts and approaches for text indexing within and across videos.

- In the domain of student presentation videos:
  - Approaches for visual indexing via "fuzzy" segmentation of scenes. These approaches are based on detection of abrupt and gradual visual changes, which are generally suitable for unstructured videos.
  - Methods for speaker indexing, including speaker segmentation and clustering, and face indices for content browsing.
  - Extensive user studies conducted with over 1,000 students in a three-year period. These user studies drove the design of the user interface and changes
to content indexing approaches. Through analysis, we understand which indices help in fast and successful completion of targeted search tasks. Generally, structural cues significantly improve completion, while the availability of actual video in addition to extracted index cues negatively impacts search tasks.

- In the domain of unstructured videos:
  - Approaches for an interactive user interface for video browsing, which provides the user with various controls of organizing visual and textual indices.
  - Approaches for keyword and key phrase filtering, ranking, and visualization for text browsing.
  - Methods for annotating, and creating a discussion-like forum for video content. Annotations are added to specific locations in the video instead of tagging the entire video.

- In the domain of automatic speech transcription: An approach for text-to-speech alignment, which is necessary to correctly map searchable text cues to their original speech. This approach is applicable when ASR systems do not provide temporal information, or when manually generated transcripts require alignment to speech.

- In the domain of video collection organization: a novel taxonomy browser, which models the two dimensions of temporal and semantic similarity of video categories via a three-dimensional cylinder user interface.
In the domain of computer human interaction: studies to understand user interaction with standard video players versus index-cue driven players. We identify two video browsing approaches, audio-visual and visual-only browsing. We also demonstrate the added benefit of rich indices for searching and browsing of video content.

In Chapter 4 we discuss our early work on visual and textual indexing, and user interfaces for lecture videos.

Visual indices (Chapter 4.1) are automatically generated by classifying video keyframes and clustering them into teaching units of similar contents. We define the term keyframe for unstructured videos as an image representative of the contents of a segmented video scene. This work is motivated by improving a conventional lecture browser with a linear keyframe list of 200-350 images per lecture. We first classify keyframes by their original source in the classroom: blackboard, hand-drawn sheets, contents from the computer, printed media illustrations, shots of the instructor, and shots of the students. Due to the high degree of redundancy in imagery from the slowly evolving lectures, we then cluster keyframes of type blackboard and hand-drawn sheet into teaching units of similar written contents. Classification and clustering effectively create a higher-level semantic for individual lecture videos. This work is also discussed in condensed form in [Haubold and Kender, 2003].

We explore textual indices (Chapter 4.2) to establish cross-lecture cues, which are useful for browsing video content for a typical university course of 40 hours of audio-
We introduce approaches for filtering and creating browsable visualizations for highly inaccurate automatically generated transcripts ($\approx 75\%$ Word Error Rate). We show that it is possible to map inaccurate lecture transcripts to their contextually matching textbook chapters with a reasonable accuracy of more than 70%. This work is also discussed in condensed form in [Haubold and Kender, 2004].

In Chapters 4.3 we demonstrate through user study evaluation that lecture browsers with meaningful visual cues are helpful for fast content retrieval and are well-received by students. We further establish in Chapter 4.4 that the availability of audiovisual resources for study through our video browsing platform VAST MM (Video Audio Structure Text MultiMedia) positively impacts exam scores. Our study is based on making available lecture video resources only during the final exam period for two selected computer science courses. We compare the improvement in midterm to final exam scores between two groups: one which voluntarily opted out of using the resource and one which actively used the resource. Students who actively participated experienced an average improvement of one third of a standard deviation in exam score.

Following a popular trend of using electronic presentation media as the predominant teaching tool, we investigate more general issues of video browsing, which are applicable to lecture videos and student presentation videos (Chapter 5). This work is also discussed in condensed form in [Haubold and Kender, 2005; Haubold and Kender, 2007a; and Haubold and Kender, 2007c].

We investigate visual segmentation (Chapter 5.1) as a function of visual changes drawn from a variety of indicators sensitive to the relatively slowly evolving and
uneventful video content. Unlike for highly edited videos, visual changes in unedited videos are defined as gradual changes due to camera actions, such as zooming or panning, gradual changes in contents, such as a person moving into the camera view, or abrupt changes, such as the progression through electronic presentation slides. However, any variability of a visual change in unedited videos cannot be easily categorized as important/unimportant. We therefore include its threshold as a free parameter, which can be manipulated in the user interface.

Audio indices (Chapter 5.2) are determined by several interrelated measures, including speaker segmentation, clustering, and face detection. Audio changes in unstructured video are mainly derived from switching speakers, such as the discourse between an instructor and the class or the successive parts of a presentation by several students. The audio track is segmented by speaker (Chapter 5.2.1), drawing from our observation that videos with multiple speakers offer interesting actor-oriented cues beyond keyframes and speech. In particular in presentation videos the viewer is not only interested in “what” is presented, but also “who” is presenting. Speaker segments are separately emphasized in the video browser and would eventually be used to determine recurring instances of the same speaker through speaker clustering (Chapter 5.2.2) and, in combination with face detection, be used to create an automatic face index of a video (Chapter 5.2.3). Our work on speaker clustering is also discussed in condensed form in [Haubold and Kender, 2008].

Textual cues for presentation videos (Chapter 5.3) are created similar to those of lecture videos (Chapter 4.2). In addition to applying these cues for cross-video indexing,
keywords and phrases are also visually mapped to their original speech. We include a user control in the video browser, which interactively clusters keywords and key phrases based on recurrence. This tool, developed initially for textual indices for lecture browsers (Chapter 4.2), emphasizes the difference between topical and thematic text.

In Chapter 5.4 we present evaluation for extensive user studies performed with over 1,000 users in a period of three years. Quantitative analysis of various targeted tasks and anecdotal feedback from students drove modifications and additions to the VAST MM indexing and browsing system. We present compelling results on the positive effect of structural cues and the negative effect from availability of actual video for search and retrieval tasks. Through evaluation of a difficult search task from each of five consecutive semesters we show that the improvements to VAST MM have lead to a significant decrease in average required time from 436 seconds to 128 seconds and an increase in average successful completion rate from 57% to 97%.

Chapter 6 outlines four common indices and tools that are applicable to all unstructured videos and video browsing interfaces.

For lack of information from popular automatic speech recognition software, we investigate automatic text-to-speech alignment in Chapter 6.1. We leverage the relative ease with which a subset of phonemes (monophthongs and fricatives) can be recognized, and perform phoneme-level matching between text and speech. This process is generally useful for temporal alignment, whether for automatically or manually generated transcripts. This work is also discussed in condensed form in [Haubold and Kender, 2007b].
In Chapter 6.2, we introduce methods for extracting phrases from transcripts and ranking words and phrases according to their semantic saliency. Ranked text indices are used during text search to place different weights on query tokens. Text ranking is also applied for the keyword and key phrase user interface in the video browser to differently emphasize unique versus common terms. This work is also discussed in condensed form in [Haubold, 2007].

We investigate video annotations and bookmarks for videos on Chapter 6.3. We found a notable discrepancy in the common webpage-style usage of discussion threads and bookmarks for videos. Instead of annotating specific content in videos, present tools treat videos as isolated objects, similar to images. Unlike images, however, videos contain a temporal dimension, which for unstructured videos can extend up to several hours (e.g. a 2½ hour lecture). Because effective annotations are time-dependent, we have decided to provide private bookmark and publically viewable annotation features sensitive to specific locations within the video. Such features make it possible to focus a discussion on video content instead of only the video.

In Chapter 6.4, we introduce a novel taxonomy browser intended for navigation of video categories. With the development of VAST MM over three years, the number of videos in the database grew by an order each year (2005: \( \approx 10 \), 2006: \( \approx 100 \), 2007: \( \approx 1000 \)). While search tools are useful in finding specific material, they do not necessarily replace tools for browsing categorically related videos. We have designed a three-dimensional cylinder UI, whose linear dimension (cylinder sections) maps to time intervals, and whose spherical dimension (the surface area) maps to a space used to
represent semantic similarity between category labels (drawn on the surface). Categories spanning several time intervals are visibly connected by lines, making it easier to track them over time. The 3D cylinder can be freely manipulated by panning and rolling via mouse movements.

Finally, we provide a comparative study of user interaction between standard video players and index-cue driven browsers like VAST MM (Chapter 7). We have designed nine search tasks of varying difficulty over a video database of 170 hours of video. One group of users completed the study with a standard video player, while the other group used the VAST MM browser. As anticipated, the time required for search without index cues is inversely proportional to matching video content, whereas it is approximately constant when index cues are available. Through analysis of interaction with standard video players we also determine that users generally adhere to one of two browsing styles: audio-visual browsing or visual-only browsing. Such observations should drive future work on video player implementations.
Chapter 2

Prior Work

Personal interaction videos are rich in content but typically lack frequent action events, which are commonly found in genres of news, sports, and film. They are to the most part unedited and often contain long sequences during which a single topic is covered. Determining structure of their contents relies on approaches of content analysis for determining contextually coherent units, and their temporally recurring instances.

2.1 Segmentation of Video

Segmentation of video into shots tends to be based on low-level features, such as histogram changes, MPEG motion vectors, Gabor energies, textures, etc. The Cornell Lecture Browser [Mukhopadhyay and Smith, 1999] uses histograms to detect presentation slide changes; Smith and Yang use it to detect cuts in news and other non-presentation videos [Smith and Kanade, 1998; Yang et. al., 2003]. Feature vectors from such low-level features are also used for statistical approaches to segmentation, and
machine learning methods of classification of shots. Souvannavong applies Latent Semantic Indexing (LSI), a well-known method in text analysis [Landauer et. al., 1998], to clustering of shots in news videos [Souvannavong et. al., 2004]. Dorai uses low-level feature vectors to train classifiers for video shot types, such as blackboard/whiteboard, narrator, and slide text, among others [Dorai et. al, 2003].

### 2.2 Segmentation of Audio

Segmentation and classification of audio is approached similar to video. Low-level features include volume, zero-cross rate, frequency centroids, Fourier analysis, and the widely used Mel Frequency Cepstral Coefficients (MFCC), among others. Fujii uses the audio signal of a lecture video to detect pauses in speech, which are then used to model text-from-speech topics [Fujii et. al., 2003]. Low-level audio features are also the basis of video classification into sports, news, and commercials [Liu et. al, 1998]. The Cornell Lecture Browser system artificially introduces audio signals as cues for synchronization between several cameras. Chen introduces a robust speaker segmentation algorithm using low-level MFCC features and the Bayesian Information Criterion [Chen and Gopalakrishnan, 1998].

The most useful information derived from the audio stream is the text transcript, captured through Automatic Speech Recognition (ASR). It has been applied ubiquitously in video analysis, for text query and search is a better-understood problem than its visual analog. Fujii applies ASR to generate text transcripts for lecture videos, and Waibel for meeting videos [Waibel et. al., 2003] to serve as searchable and browseable indices. Smith and Yang apply transcripts from news videos as indices for searching and
querying. In some work, transcriptions serve specifically as text indices to audio archives, for example in video mail [Young et. al., 1997], or news streams [Whittaker et. al., 1999]. While ASR algorithms use machine learning to improve accuracy given speaker and language models, their results are not perfect. When no models are customized, accuracy drops dramatically [Witbrock and Hauptmann, 1997]. In the case of lecture videos, analysis of content and training of language models can lead to better accuracy [Glass et. al., 2004].

2.3 Segmentation of Text

Text is also used as a separate medium in video analysis for contextual segmentation and clustering purposes. Typically, statistical methods are applied to select interesting and useful words and phrases, while discarding topologically redundant ones [Yang and Wilbur, 1996]. Common approaches are the application of text frequency, inverse document frequency (TF-IDF), and LSI. Lin and Ponceleon show how words from lecture video transcripts are used as low-level features to detect topic changes, and thereby segment the video [Lin et. al., 2004; Ponceleon and Srinivasan, 2001]. Lin and Yang analyze word classes to rank and form better comparisons. Yang also introduces external corpora, such as WordNet [Fellbaum et. al., 1998] and news web pages, to expand queries.

2.4 Segmentation of Events

Imagery, audio, and text are the most commonly identified media in video. Their analysis depends mostly on video-internal content that can be extracted with relative
ease. Research by Waibel suggests that environmental cues, such as phone rings or door
knocks in a meeting environment, can be applied to structure video content. However,
event detection is beyond the scope of video content analysis as defined here.

2.5 User Interfaces

The work of segmentation, indexing, and structuring of video is eventually
presented in user interfaces, whose primary goal is to allow for parallel data exploration
of an inherently serial medium. Currently available media players still rely to the most
part on a time slider and fast forward/rewind functions. An efficient interface attempts to
display audio, video, and textual information as compactly as possible, allowing a viewer
to advance to any part of the video, while playing back as little as possible. Literature
provides many examples of prototypes for visual and audio summarization interfaces and
browsers. Lee presents an overview of video browsing issues and a comparison of 15
prototypes from research [Lee et. al., 1999]. Some of the most important features
mentioned are abstraction of information, and the Visual Information-Seeking Mantra:
Overview first, zoom and filter, then details-on-demand [Shneiderman, 1996]. Li
evaluates features for video browsers for individual genres of lecture, presentation,
entertainment, and other videos [Li et. al., 2000]. Specifically for lecture videos, users
preferred using tables of content to skimming the videos.

Most interfaces use keyframes as means of visual summaries, such as the
keyframe-based UI for digital video [Girgensohn et. al., 2001] (see Figure 2.4), where a
space-optimizing mural of differently sized keyframes is used to present a video segment.
Size of keyframes is proportional to importance as determined during segmentation and
indexing. Earlier work on lecture videos used a popular web page framing approach, where video playback features are separated from electronic slides, and text indices. The Cornell Lecture Browser and Classroom 2000 [Abowd et. al., 1996] (see Figure 2.2) are a few such examples. Some more recent interfaces cluster keyframes based on feature vectors [Worring et. al., 2004; Tang and Kender, 2006] (see Figure 2.3), allowing for zoom on dense clusters for detailed exploration. Altman proposes a semantic lecture browser (see Figure 2.1), which places emphasis on pedagogical events, such as in-class discussion, theorem, equation, diagram, example, etc. [Altman et. al., 2002]. The interface is based on a hyperbolic graph of events from a lecture, and allows navigation to related events. However, the interface remains a suggestion, as no related work on segmentation and video content understanding exists.

While video summaries are based on their visual video source, browsing audio archives requires a multi-modal shift: it is possible to very quickly skim text, but not serial audio streams. ASR is used to create that shift into the visual mode of text. SCAN for news audio [Whittaker et. al., 1999] (see Figure 2.5) and the Video Mail Browser [Young et. al., 1997] (see Figure 2.6) show user interface examples for skimming and querying of audio streams using ASR transcribed text as a medium.
Figure 2.1. Semantic Exploration of Lectures [Altman et. al., 2002].

Figure 2.2. Classroom 2000 [Abowd et. al., 1996].

Figure 2.3. Video Archive Access [Worring et. al., 2004].

Figure 2.4. Keyframe-Based UI for Digital Video [Girgensohn et. al., 2001].
2.6 Capture of Structure

Several researchers have investigated videos with respect to extraction of structure. They are addressing the need to build hierarchies around serial content, so that shots are no longer self-contained contextual entities, but can be linked and made relevant with other content. Conceptualization of content discussed is one such approach in which video shots in the news domain are annotated with trained concepts descriptive of the scene [Natsev et. al., 2004; Kender and Naphade, 2005]. The resulting concepts for shots can be used to track news episodes over time, or can be searched with additional query expansion given dictionaries such as WordNet, demonstrated by Haubold on TREC 2005 Video data [Haubold et. al., 2006]. Hauptmann takes a similar approach for various genres of video, including promotional and documentaries [Hauptmann et. al., 2003].
Sources of annotation include Optical Character Recognition (OCR), ASR, face detection, and image comparisons for querying. Sundaram investigates structure and hierarchy of scenes in film derived from domain-specific features, such as dialogue and cinematographic rules [Sundaram and Chang, 2000].

Video summarization and indexing approaches have been studied in a variety of genres, including lecture and presentation videos. Analysis and domain-specific methods of content extraction have lead to customized user interfaces that emphasize features in those domains. Structure in videos, however, has been sparsely addressed, in particular in a multi-modal space covering video, audio, and text. Tables of content and indices for books are based on structure of content. We propose that a similar framework should be applied to video content.
Chapter 3

VAST MM: An Orientation

This chapter gives an overview of the Video Audio Structure Text MultiMedia (VAST MM) indexer and browser, with motivation and evolutionary history.

VAST MM is one of the first multimedia indexers and browsers for large (> 100 hours) collections of unstructured video (see Appendix B for the extent of our video database). It makes available multimodal cues from audio, visual, and textual domains in an interface, in which users can interactively vary the amount of information displayed. Because of the nature of unedited video, indexing approaches must be sensitive to their highly varying content. Meaningful cues are drawn from events, such as audio and visual changes, and from extracted and inferred content, such as imagery and speech. We apply methods of segmentation and indexing by both audio and visual data.
3.1 Process

Video contains several dimensions of data in the visual and audio domains, each providing several approaches to content analysis. The visual domain contains human-identifiable cues, such as faces, text, diagrams, objects, motion, among others, all of which partake in scenes and events. The audio domain contains cues from speech (text), unique voice prints, emotion through intonation or volume, among others. Analysis and content extraction must necessarily occur over whatever subset of these modalities best represents the content for search and browsing. For presentation videos made available through VAST MM, the set of audio-visual cues includes keyframes from scenes, faces from speakers, speaker-based segmentation, transcripts, and filtered key phrases.

Visual segmentation is performed by combining two methods that determine scene changes, one for abrupt and one for gradually changing content. Speaker segmentation determines individual student appearances, and is the foundation for extracting headshots for the visual speaker index. Automatic transcripts are generated, but without speaker or language models, due to the significant additional burden such an endeavor would present. Instead we apply approaches for filtering the highly inaccurate transcripts to produce relevant keywords and phrases. An outline of the analysis is presented in Figure 3.1.
3.2 Why it is Necessary

Standard video players, whether on-line or off-line, offer little beyond player controls of play, pause, stop, and a timeline-based location slider. While highly edited entertainment media tends to be played back by their audience without the need for more advanced features, videos in other genres, like presentation or instructional videos require tools for retrieval of specific content. For example, a typical university lecture encompasses more than 40 hours of video material, which is an unwieldy amount of information to search through for a short discussion. Audio, visual, and textual cues are necessary to provide searchable and browsable indices for larger video databases; without such cues, video material remains a cumbersome medium to use compared to the text-driven WWW.
To demonstrate the benefit of browsable and searchable cues, we compare user interaction between standard video players and index-cue driven browsers for search tasks of varying difficulty using a large database of more than 170 hours of video material. Difficulty here is defined as the amount of matching video contents for a given search task. Through user studies, we identify two principal user interaction methods when cues are not available (see Chapter 7.2). Visual-only browsing entails the use of the timeline location slider to visually skim video keyframes; most users view a keyframe in less than one second, and then skip less than three minutes of contents to the next keyframe. Audio-visual browsing entails watching and listening to video contents; most users view less than 10 seconds of video, then skip less than 8 minutes to the next clip. We find that users strictly adhere to one of these two browsing methods, both of which occur with the same prevalence rate. We also find that audio-visual browsing tends to require less time for the average search task than visual-only browsing (531 seconds versus 846 seconds).

In comparison to index-cue driven browsers featuring text search and browsable audio, visual, and textual cues, standard video players clearly disappoint in their ability to help the user find material quickly. In our user studies, we determine that users complete significantly more tasks in a much shorter time frame with an index-cue driven browser such as VAST MM. Overall completion rates exceed 70% as compared to 33% and average required time of 110 seconds is also significantly lower than 646 seconds for standard video players. Finally we show that in an index-cue driven browser environment, search time is nearly constant regardless of the difficulty of the task,
whereas with standard video players, search time is inversely proportional to amount of
matching contents.

Clearly the need exists to establish video browsers with a variety of multi-modal
cues to better represent the otherwise overwhelming amount of information stored in
videos. VAST MM illustrates what types of cues work well and which ones do not in the
domain of unstructured videos.

3.3 Evolution in Pictures

The VAST MM content indexer and browser have undergone various changes
and additions, a result of three years of user studies and evaluation. Fundamental visual,
audio, and text cues were retained throughout all versions; however, they were
consistently refined to better represent the underlying video content.

In the first version of VAST MM (see Figure 3.2), keyframes were extracted for
each scene, subject to segmentation on a simple frame-by-frame motion metric. This
metric worked particularly well on the small dataset of 8 hours of video shot without
variability in zoom. The audio track was segmented by speaker, and speech-to-text
transcripts were filtered by an expected corpus of keywords and phrases. ASR transcripts
without temporal alignment were approximately aligned using a linear interpolation
algorithm. The user interface provided a temporally zoomable summary with the
extracted static cues. In its first trial, a portrait style layout was quickly dismissed by
users as unintuitive and difficult to navigate; the landscape version now prevails.
Keyframe thumbnails could be enlarged to their high quality version by mouse clicks.
Figure 3.2. VAST MM Browser Version 1. User Interface for video/audio segmentation and text augmentation. Videos are selected from the list on the left side. Actual video (in this version of VAST MM, stored on the local hard drive) appears in the lower left frame. Video summaries in the right frame appear on a segmented linear timeline, whose scale can be set between 1 and 100 video frames/pixel. The row of thumbnails provides access to high-quality keyframes. The timeline presents a combined segmentation of audio and video. The green row shows audio activity, the red row video activity, and the yellow rows displays index and content phrases.

The UI also featured a video player based on the Java Media Framework (JMF). However, JMF lacked a working implementation for streaming capabilities. User studies were therefore arranged in a setting with local copies of videos.

The second version of VAST MM (see Figure 3.3) addressed the problems observed in version 1, while adding new features for information retrieval. With a larger set of videos exhibiting a wider range of recording characteristics, visual segmentation by
Figure 3.3. VAST MM Browser Version 2. User Interface featuring video summaries in the multi-modal domain. The video summary is displayed as a collection of horizontal tracks, each representing a different modality: thumbnail images, time line, speaker segmentation, visual segmentation, search phrases, topic phrases, and content phrases. Three parameters of the summary can be tuned: (1) scene segmentation (coarse versus fine), (2) zoom (overview versus detail), and (3) text context (emphasize repetitive phrases within a time interval). A separate visual face index links actors to the segments in which they present. Faces were manually extracted for eight videos and evaluated in a user study. A video and a keyframe player (in this version of VAST MM, video is streamed on-demand from a server) are available as well, but are not emphasized in the user interface.

Frame-by-frame motion was insufficient, and yielded unsatisfactory results on videos with zoom and pan activity. The new segmentation approach still measures frame-by-frame activity to detect abrupt changes, but also computes changes occurring between distant frames to account for motion from camera activity and motion in the scene.
Transcripts were more accurately aligned using phoneme matching, and filtered phrases were assigned measures of saliency.

The user interface was remodeled to include more interactive features to customize the amount of information presented. A scene segmentation slider was added to change the granularity of pre-determined scene cuts, which implicitly adjusts number of thumbnails displayed. Filtered text cues were visually emphasized by their semantic rank, placing emphasis on more descriptive terms. A separate slider adjusted a text grouping parameter to easily filter out recurring themes. Separate text search engines over metadata, filtered phrases, and raw transcripts were provided, although at this point they acted only as a filter of the long video list and did not specify matching locations in the video. The earlier JMF video player was replaced by a custom implementation of an MPEG1 stream player. Also introduced was a keyframe player, which played back visual information in high speed. For user study purposes, this version of VAST MM also featured a face index, a popular approach to summarizing actors in a video. Two differently oriented faces were extracted for each speaker segment in a video and included in the summary.

The final version of VAST MM (see Figure 3.4) features two new tools for organization of larger collections of video, motivated by the growing collection of over 1,000 videos. A 3D cylindrical category browser (introduced in Chapter 6.4) presents video categories in a temporal and similarity-based layout that emphasizes contextually related categories while mapping them over the time period in which they are relevant.
Figure 3.4. VAST MM Browser Version 3. Users can browse categories; search and view videos and their summaries; create public annotations and personal bookmark. Shown in this view is the browsable video summary for one video. The face index, while found to be an important visualization, was not incorporated here, because automatic extraction of facial data is an ongoing research project.

The second tool, a text-based video comparison tool, visually emphasizes recurring phrases and can be used to determine themes across videos. A single text search engine now answers queries by a weighted fusion across metadata, filtered phrases, and raw text. Search results are also identified in the summary instead of acting only as a video filter. Finally a visual concept annotation track, part of a video’s browseable summary is included, which may find future application for annotating individual keyframes with visual concepts.
Figure 3.5 summarizes additions and changes to content indexing and user interface in VAST MM over its development time.
Chapter 4

Lecture Videos

This chapter introduces our early work on instructional video indexing and user interfaces. It eventually inspired research on unstructured videos, which lecture videos are a subset of. This chapter also presents user studies completed with the early lecture video browser and the more recent VAST MM system.

Lecture videos are a popular medium for lecture dissemination originally intended for distant learning programs and cumbersomely delivered in hard-copy VHS format. The emergence of high-speed networks and ease of digital media production and distribution lead to significant technical improvements for distance education while also increasing popularity of recorded lectures among on-campus students. Even though the mode of delivery changed, little about content playback and browsing improved. Common digital media players merely augment traditional functions of play/pause/stop with a location slider. Visual, audio, and textual cues for information retrieval are lacking entirely unless they have been semi-manually added by the content producer. Our work on lecture videos
focuses on automatically generating such cues for browsing and searching, taking advantage of characteristic features of this video genre.

With its highly controlled setting and predictable events, the lecture videos we use are shot nearly unintrusively by a semi-professional camera operator using professional video equipment specifically calibrated for the video-enabled classroom. Two to three cameras suffice in covering the most important areas: a front-facing camera captures the instructor, podium, and blackboard/whiteboard, an overhead camera captures written material and illustrations on the podium, and an optional side-mounted camera selectively captures the podium and the class. A separate screen capturer records material displayed on the instructor’s computer, including web pages, electronic presentation slides, etc. A directed wireless microphone worn by the instructor captures speech, while an optional omni-directional array of microphones controlled by the camera operator is able to pick up audio from the classroom when students interact with the instructor. In general, the quality of lecture videos is sufficient to follow the material both visually and audibly. The most influential parameters affecting the audio-visual delivery are spatial and data resolution. In the research presented here, we exclusively work with low resolution compressed video material.

4.1 Visual Indices

Video segmentation, indexing, and visualization are essential parts of content-based video retrieval. A successful system would allow the user to review even a long video (up to 150 minutes) by means of some visual summary, and would provide ways to quickly retrieve video contents based on useful clustering and user interface
visualizations. Characteristically of their genre, instructional videos tend to be taken in a
set environment with a small set of well-defined areas of interest. The segmentation
process should exploit this underlying structure.

Research in the area of segmentation and visualization of instructional videos is
still in its early stages. Most related work has focused on indexing methods for news
videos [Ichiro et. al., 2001, Zhang et. al., 1995], sports videos [Zhong and Chang, 2001],
and situation comedies [Aner and Kender, 2001]. Some work has been done on the
segmentation of the blackboard frames of instructional videos taken in a specially
instrumented classroom [Onishi et. al., 2000]. However, many instructional videos
contain material from sources other than the blackboard, and from environments not
specifically designed for video analysis.

4.1.1 Overview

The summarization and indexing of a video begins by collecting keyframe images
taken at points of substantial change. In our experimental application, consisting of 17
videos of long lectures, these keyframes were chosen by a proprietary software system
developed for the Columbia Video Network (CVN), which is sensitive to image motion.
On average, a keyframe is produced every 20 to 25 seconds, so 75 minute lectures
contain about 200 keyframes, and 150 minute lectures about 350. However, even casual
users of these keyframes have noted that they tend to be rather repetitive, as much
lecturing consists in emphasizing verbally what has been visually created. By grouping
together keyframes of similar contents into topic clusters, the complexity of the keyframe
set can be reduced by 80 to 95%.
Figure 4.1. Lecture Video Browser: Topological index with keyframe summary. Each keyframe media type is assigned a distinguishable color as well as a descriptive icon. Vertical keyframe summaries are aligned with media type icons; horizontal topological groupings capture topic commonalities. Icons and keyframes are clickable; they select topics, magnify the thumbnails, and pop up the video at the appropriate frame.

Structured experiments involving the responses of 11 students and one instructor to three alternative designs yielded a clearer idea of what kind of segmentation, indexing, and visualization would be most useful. The final working design consists of two separate graphs to display the same data by different means (see Figure 4.1): an abstracted Topological View displays the media type and relative (not absolute) temporal location and relationship of topics in the video; a thumbnailed Keyframe View facilitates access to full-size keyframes (and the video itself) within each topic. Since keyframes from a given topic can appear at any point in the lecture, temporal discontinuities within a topic are
Figure 4.2. Lecture Video Browser. White vertical bars highlight a selected topic in Keyframe View. Magnified semi-transparent images appear above the thumbnailed keyframes, while the keyframe summary is browsed with the mouse cursor; a yellow row and column mark the selected thumbnail. Upon clicking on a thumbnail, the full-sized keyframe appears in the Image Frame, and the video can be played back in a web browser starting at that keyframe. Keyframes from a selected topic can also be played back continuously with the Keyframe Player.

illustrated by tapering connecting lines, as seen in Figure 4.1. Keyframes are further distinguished by their media types. We have identified six: “board” for shots with a majority of blackboard (or whiteboard) content; “class” for shots of students in the classroom, e.g. during instructor-student interaction; “computer” for keyframes capturing the computer screen; “illustration” for printed material captured on the podium; “podium” for shots of the front of the classroom, possibly with the instructor but excluding detailed blackboard shots; and “sheet” for handwritten material shot on the podium, an alternative to using the blackboard.
These two user interfaces allow the user to search and browse for specific parts of an instructional video, and they also highlight potentially important portions of a lecture. The Topological View automatically lays out interrupted topics in a visually non-interfering planar array; this graphically captures those semantically dense points in the lecture with interaction between different topics. The Keyframe View enables the user to retrieve full-sized keyframes in a separate window by sliding the mouse over the thumbnailied images. First (last) keyframes for a given topic can be determined by finding the start (end) of the topic using icons in Topological View; clicking on these icons then highlights all the related keyframes in the thumbnail strip below, regardless of their temporal separation (see Figure 4.2).

4.1.2 Media Type Classification: First Pass

The first step in the segmentation process is to assign each keyframe to one of five media types. (Because one media type, “class”, occurs so rarely, we have omitted it.) This classification first uses a decision tree of static image feature filters (see Figure 4.3), such as overall color information, color information in certain spatial arrangements, color patterns, and features such as edge information. A post-processing dynamic filter stage follows.

In our application, the proprietary keyframe detector is aware of two special sources of imagery: the control room-generated title keyframes, and PowerPoint keyframes. These are stored with distinguished names and are easy to recognize and classify.
Keyframes are first analyzed for characteristic visual features intrinsic to computer-generated material. Empirically it is observed that computer screen images, when scaled to fit the video frame, are padded with a border of black pixels from 5 to 10% of the image. Additionally, because of the need for high ambient lighting for good visual images of the instructor, board, or students, computer keyframes are among the darkest keyframes. Classifying black bordered or dark frames as computer frames was found to be 100% accurate.

Remaining keyframes (which may include some additional computer frames) are analyzed by color content. Keyframes with mostly green color are labeled as candidates
for media types board and podium; those with mostly white color are candidates for computer and sheet; and any remaining are candidates for a more complex analysis for four media types.

The semantic difference between the board and podium media types is derived from the behavior of the instructor. When the instructor is using the blackboard, the camera operator tends to focus on the blackboard, to capture blackboard content. If the instructor is interacting with the class, the camera tends to focus on the instructor, to capture gestures and facial expressions. Classification can therefore be based on the spatial arrangement of the blackboard color relative to the instructor.

Podium keyframes contain portions of green color concentrated in vertically central regions of the image, with bottom portions of the image colored differently (see Figure 4.4b). Predominantly green keyframes lacking green color in their bottom 10% are classified as podium. Although the distinction between podium keyframes and board keyframes is not crisp, any errors are not critical; the primary consequence is that podium keyframes are simply not clustered into topics of similar visual content, a process performed only for keyframes of type “sheet” and “board”.

Board keyframes empirically are observed to be of two major kinds. The first is a large green area that includes a green lower border. The second is a smaller green area (due to occlusion by the instructor) that nevertheless has a nearly complete green border on all sides.
Board and podium keyframes are accurately classified at this stage 97% of the time. Errors consist of predominantly green computer or illustration keyframes; some of these are detected and corrected in a post-processing phase.

Candidates for the computer media type, which at this stage are primarily white, are distinguished by the presence of the horizontal linearity of their contents: rectangular imagery, tables, menu bars, etc. By using a Laplacian edge detector, we extract an edge image, and compute from it a weighted measure of the presence of horizontal lines, giving exponentially more weight to longer lines. Keyframes with a measure above a...
threshold are classified as computer frames, with 99.9% accuracy. The two frames classified in error were due to the rapid motion of the instructor’s pen; this motion generated the digitization interlacing artifact of every other scan line being dark, giving the appearance of horizontal lines.

Remaining (and still primarily white) candidates are classified as sheet media type if they include a sufficiently large amount of white, light gray, and/or skin tones, with 100% accuracy.

Keyframes without dominant green or white regions can still be classified as any media type other than sheet. This last stage of processing handles, for example, zoomed-out frames of the board or podium, or computer frames and illustrations having colors other than white or green. An empirically derived sequence of tests revisits the computer, podium, and board media types, leaving the illustration media type to be the default classification if the other three types fail.

The keyframe is first tested for media type computer by computing its horizontal line measure, and a related measure of vertical or horizontal color repetition. Computer frames are not affected by classroom ambient lighting conditions, and are therefore characterized by long vertical or horizontal runs of very similar pixel values. Frames exceeding thresholds in either measure are classified as computer. The majority of classification errors occur at this point; these tests appear to be overly general, although all remaining computer frames are classified 100% accurately.

Frames not meeting these conditions are re-examined for the heuristic features specified for board and podium with similar classification accuracy.
4.1.3 Media Type Classification: Second Pass

The preceding heuristics are based on static image features. Two post-processing methods exploit temporal relationships to verify and refine the media type classification.

The first post-processing method focuses on the small number of fade-out keyframes at the end of a video that mirror the fade-in at its beginning. The video begins and ends with the same black panel with white lettering that identifies the lecture and instructor, generated by the control room’s computer. They should be classified as podium frames, but may have been misclassified, depending on the content they have been dissolved with. These frames are therefore reexamined for large dark areas, in a temporally reverse manner (see Figure 4.5).

A second post-processing method notes that a sequence of computer frames might include keyframes with predominantly green color (e.g. default Windows NT background); these may have been misclassified as board or podium. The beginning and
end of a computer subsequence are easily and reliably detected. The color of those keyframes classified as board or podium frames between these endpoints are compared to their computer frame neighbors, and reclassified as computer frames if similar enough. This heuristic corrects about $\frac{3}{5}$ of these errors.

### 4.1.4 Topological Segmentation

The proprietary software that selects keyframes does so in way that is mostly sensitive to instructor motion, concentrating its captures during periods of relative visual calm. Consequently, the keyframes are highly redundant. Nevertheless, these frames provide a good measure on how information in the classroom is developed and altered, and illuminate differences in teaching styles: some instructors tend to distribute information linearly, while others like to jump back and forth between topics.

In order to retain these characteristics while reducing redundancy, we cluster similar keyframes into topics based on a keyframe’s visual content. Two keyframes are clustered together if the more recent frame elaborates on the visual information found in the more distant one. This clustering reduces a large number of keyframes to a set of clusters (topics) of similar keyframes that is 5 to 20% of that size. An example is given in Figure 4.6. Assigning a letter to each keyframe denoting its topic, a long string of similarly labeled frames results.

We apply topological clustering only to board and sheet keyframes, because these are the most informative media types. (It is unclear how computer, podium, or class frames can be clustered according to visual contents: computer frames are often not
Figure 4.6. Topological Clustering. For the video displayed in Figure 4.1, 323 keyframes have been reduced to 41 clusters. X and Y denote podium and computer keyframes, respectively. Letters A through K denote distinct clusters of board keyframes. The exponent for each letter denotes how many frames of the same topic appear in each contiguous sequence.

effused but simply superceded, and podium and class frames have no distinctive visual content.) The clustering of board and sheet keyframes is broken into two steps. First, images are filtered to extract writing. Secondly, images of this writing are matched using selected sub-windows of content.

4.1.4.1 Filtering

The essential distinctions between media types have an operational consequence: different sets of filters are applied to either type. For board keyframes, this means filtering out the board and all surrounding artifacts, while sharpening the chalk marks. For sheet keyframes, a filter removes all areas that do not relate to the material written on the lighter background.

The board filter addresses the problem of poor board versus chalk contrast; Figure 4.7 shows an example of low contrast. The board filter explicitly encodes algorithms to separately extract the background and the foreground. Each step is represented by a branch of the flow diagram in Figure 4.7: background in (b) through (e); foreground in (f) through (i); their merger in (j) and (k).
Figure 4.7. Filters for Media Type “Board”. Above: flow diagram of filtering process. (a) original image, (b) green color filter, (c) flooded board, (d) outline of flooded area, (e) complete flooding of outlined area (board), (f) edge filter, (g) horizontal and vertical color similarity filter, (h,i) morphological filters, (j) ANDed combination of (e) and (i) results in extraction of board contents, (k) large pixel blob filter removes useless features; Below: example of video frame (left) and result of filtering (right).

For board frames, the background is extracted in 4 steps. First, potential blackboard pixels are isolated with a simple green color filter (see Figure 4.7(b)). Using the edges found by the edge filter in Figure 4.7(f) and the potential blackboard pixels from Figure 4.7(b), the board in Figure 4.7(a) is flooded to obtain the largest closed blackboard region(s) in Figure 4.7(c). Because the result excludes any writing, the result is outlined in Figure 4.7(d) and flooded again in Figure 4.7(e).

The foreground is extracted beginning with a 3x3 Laplacian edge filter in Figure 4.7(f). Edge artifacts from the borders of homogeneous regions are detected by a horizontal and vertical color similarity filter and removed in Figure 4.7(g).
Figure 4.8. Filters for Media Type “Sheet”: Above: flow diagram of filtering process, (a) original image, (b) white color filter, (c) flooded sheet, (d) outline of flooded area, (e) complete flooding of outlined area (sheet), (f) edge filter, (j) ANDed combination of (e) and (f) results in extraction of sheet contents, (k) large pixel blob filter removes useless features; Below: example of input image (left) and output image (right).

morphological filter in Figure 4.7(h) removes noise while restoring content pixels mistakenly removed from Figure 4.7(f). A second morphological filter in Figure 4.7(i) restores content pixels mistakenly removed from Figure 4.7(a). The foreground pixels are ANDed with background pixels to recover writing pixels on the board. In a final step Figure 4.7(k), large blobs of writing are removed from the filtered image because they are not useful in matching. We call this final binary frame the derived content frame.

The sheet filter uses the same methodology for extracting writing from sheets of paper. However, because ink on fresh sheets of paper usually has higher contrast than chalk on erased blackboards, no noise filtering is necessary for the foreground. The background is extracted identically, using a simple white color filter in Figure 4.8(b), a flood fill in Figure 4.8(c), an outline of the sheet of paper in Figure 4.8(d), and a final
flood fill in Figure 4.8(e). The edges found in Figure 4.8(f) are then directly ANDed with the background pixels in Figure 4.8(j), and large blobs of writing are removed in Figure 4.8(k), the derived content frame.

4.1.4.2 Matching

Matching the content between two keyframes is complicated by the three degrees of freedom allowed to the otherwise fixed cameras: pan, tilt, and zoom. These introduce translation, scale, and perspective changes. Perspective is handled by an implicit para-perspective method: both frames are considered to be made up of small local windows. As few scale-invariant features are expected, scale is explicitly modeled by successively rescaling one of the pair by a range of 14 scaling factors (from 0.6 to 1.7), experimentally derived. Therefore, starting at full scale, expansions and contractions are alternated until a match is found, or the range is exhausted.

Given two keyframes $i$ and $j$, with $j$ the more recent, we extract a set of features in $i$ by means of an interest operator, and find their correspondences in $j$. If sufficient similarity exists, frame $j$ is considered to be an elaboration of the topic in frame $i$.

Features in frame $i$ are extracted in the form of up to six identically sized, wide aspect ratio sub-windows of interest (see Figure 4.9). The selection of sub-windows proceeds as follows:

1. Divide the derived content frame into three equal vertical strips. (see Figure 4.9(c))

2. For each strip, scan the wide aspect ratio interest window over a coarse grid of locations in the image. (see Figure 4.9(b,c))
3. Count the content pixels $cc$ in this window placement. If $low \leq cc \leq high$, stop and report window position. (see Figure 4.9(d))

4. Repeat 2 and 3, except scanning bottom-up.

5. If both reports are the same, keep only one.

This heuristic search reflects the empirical observations of both the units of writing and the camera motions observed in the videos. Since panning for boards dominates tilting for sheets, the search enforces and favors a horizontally balanced window acquisition, but without neglecting vertical distribution. Windows are sized so that their height roughly corresponds to two lines of text, while their width is about twice that size to reflect the lengths of average words.
Figure 4.10. Frame-Pair Matching Algorithm: (a) find interesting sub-windows in i; (b) find corresponding sub-windows in j; (c) determine image match quality; (d) compute $\sigma$ of translation vectors; (e) compute $\sigma$ of change of intra-window distances.

Empirically, it is observed that windows of interest ideally contain between 5 to 30% content pixels. This range provides enough pixels to match with, but not so many as to prohibit the creation of distinctive configurations of pixels.

Having found windows in frame $i$ (see Figure 4.10(a)), to increase match likelihood we next blur the derived content frame $j$ slightly by opening it with a 3x3 mask. We now find the best correspondences to these windows in the blurred derived content frame $j$, and compute a total match score as follows (see Figure 4.10):

1. Find the best location of each sub-window from $i$ in $j$ in the usual manner of template matching (see Figure 4.10(b)).

2. Define image match quality for each sub-window match as the amount of matching writing divided by the total amount of writing (see Figure 4.10(c)).

3. Define consistency of translation of the windows as the negative of the standard deviation of the lengths of the translation vectors for each sub-window pair (see Figure 4.10(d)).
4. Define consistency of spatial arrangement as the negative of the std. deviation of
the errors between corresponding intra-window distances (see Figure 4.10(e)).

5. Total match score is a weighted sum of the matching number of windows, image
match quality, translation consistency, and spatial arrangement consistency.

4.1.4.3 Topic Extraction

For matching all keyframes, we define a topic $T_k$ to be a temporally ordered but
possibly non-consecutive sequence of keyframes. Topics are themselves temporally
ordered by the time of their most recent frame, $f_k$. The keyframes of the video can now be
clustered into topics by having each successive keyframe either extend an existing topic
or start a new one:

1. The first keyframe of the video forms the first topic.

2. Each succeeding keyframe of the video is matched to the most recent frame of the
most recent topic.

3. If this match succeeds, the most recent topic is extended by the new frame, and
the frame becomes the most recent frame of the topic (see Figure 4.11, case 1).

4. If the match fails, the incoming keyframe is matched in sequence to the most
recent frame of the other topics, in the order of topic recency.

5. If the incoming keyframe finds a match, it extends that topic, and it becomes the
most recent frame of the topic, and the topic becomes the most recent topic (see
Figure 4.11, cases 2 and 3). If no match is found at all, the keyframe starts a new
topic, and this new topic becomes the most recent topic (see Figure 4.11, case 4).
Identified topics:  
\[ T_1 = \{ f_1, f_3, f_4 \} \]
\[ T_2 = \{ f_2, f_5, f_7 \} \]
\[ T_3 = \{ f_6, f_8 \} \]

Sequence of topics with most recent topic as last element:
\[ T = \{ T_1, T_2, T_3 \} \]

Next analyzed keyframe: \( f_9 \)

Possible cases | New sequence for \( T_k \) and \( T \)
---|---
1. \( f_9 \) matches \( T_3 \) | \[ T_3 = \{ f_6, f_8, f_9 \} \]
2. \( f_9 \) matches \( T_2 \) | \[ T_2 = \{ f_2, f_5, f_7, f_9 \} \]
3. \( f_9 \) matches \( T_1 \) | \[ T_1 = \{ f_1, f_3, f_4, f_9 \} \]
4. \( f_9 \) does not match any \( T_k \) | \[ T_4 = \{ f_9 \} \]

\[ T = \{ T_1, T_2, T_3, T_4 \} \]

Figure 4.11. Matching Heuristics. Example outlining the four cases for a match of a frame \( f \) to a topic \( T \).

Figure 4.12. Development of contents on media types “Sheet” and “Board”. (a,b,c) Sheet contents are developed strictly forward, so matching is likewise strictly forward. (d) A new topic is easily recognized by no continued sub-windows. In contrast, board contents are subject to erasure (b,c), so matching is two-way. But new topics are still easily recognized (d).

Because for media type sheet there are no erasures, the matching is performed exactly as specified in Figure 4.12, by finding sub-windows in the older frame and searching for them in the newer one. For media type board, where erasures are common
and temporal order does not guarantee increased content, the matching is generalized and can also proceed in the reverse direction as well.

4.1.5 Analysis of Cost of Topic Extraction

We now show that, under reasonable assumptions, the matching algorithm is approximately linear. We assume that the number of topics grows linearly but slowly; as seen in the actual data. More specifically, in the 17 videos the ratio of topics to total number of keyframes is small, with average $= 14.8 / 200 = 0.074$. Statistically, a match between two consecutive keyframes occurs 89% of the time, while a match between two non-consecutive keyframes occurs 3.6% of the time, and no match occurs (i.e. a new topic is formed) 7.4% of the time.

We calculate the cost of matching as follows:

The expected cost of a match at frame $f$ is composed of three terms: the expected cost of performing a match with the previous keyframe (which happens always), plus the expected cost of finding a match with a keyframe from a prior topic (which happens when a prior topic is extended), plus the expected cost of performing a match with a keyframe from all prior topics (which happens when a new topic is started):

$$\text{Match}(f) = p_{\text{exact}} * 1 + p_{\text{previous}} * \frac{O(f)}{2} + p_{\text{new topic}} * O(f)$$

Total Cost: $M(f) = \sum_{f = \text{frames}} \text{Match}(f)$

The term $\frac{O(f)}{2}$ signifies the average search space of half the topics for matching a keyframe to a continuing topic (because $\frac{O(f)}{2} \equiv O(f)$, we use the latter notation in our subsequent calculation).
Empirically, there are an average of 200 keyframes per board or sheet segmentation; the probability of matching the current topic is $178/200 = .89$; the probability of matching with a topic prior to the most recent topic is $7.2 \div 200 = 0.036$; and the probability of not matching with any previous topic is $14.8 \div 200 = 0.074$. The cost for matching 200 frames becomes:

$$M(f) = \sum_{i=1}^{200} 0.89*1 + 0.036*\frac{O(f)}{2} + 0.074*O(f)$$

Using $O(f) = f * \frac{14.8}{200} = f * 0.074$, we compute:

$$M(f) = \sum_{i=1}^{200} 0.89*1 + 0.013f + 0.0055f = \sum_{i=1}^{200} 0.89 + 0.0068f \quad M(f) = 0.89f + 0.0034f^2$$

This result is very close to the quadratic regression in Figure 4.13, where $M(f) = 0.84f + 0.0039f^2$. For most videos, the quadratic term is negligible.
Table 4.1. Confusion matrix for Board, Podium, Sheet, Illustration, Computer: number of keyframes that were incorrectly classified. Except for the 23% error of Illustration→Computer, errors are very small (less than 3%).

<table>
<thead>
<tr>
<th>Actual</th>
<th>N = 4479</th>
<th>Classified</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Board</td>
<td>Podium</td>
</tr>
<tr>
<td>Board (696)</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>Podium (431)</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Sheet (2708)</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Illustration (387)</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Computer (257)</td>
<td>0</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 4.2. Overall performance of topological clustering for 17 videos. Errors in matching include mismatches and incorrect starts of new topics.

<table>
<thead>
<tr>
<th>Keyframes</th>
<th>Keyframe Match Errors</th>
<th>Topics Actual</th>
<th>Topics Found</th>
</tr>
</thead>
<tbody>
<tr>
<td>3394</td>
<td>112</td>
<td>239</td>
<td>252</td>
</tr>
</tbody>
</table>

4.1.6 Accuracy of Clustering Results

While no attempt has been made to optimize the performance of the segmentation tool, we have collected data over 17 extended videos measuring 40 total hours that suggests several properties of the underlying processes.

Media type classification is rather robust, as most media types were correctly detected between 97 and 100% of the time (see Table 4.1). The only exception is detection of type illustration. Topological Segmentation performed equally well at a success rate of more than 96% (see Table 4.2). On a per video basis, the number of topics in error is less than one.
4.1.7 Conclusion

Visual indices from keyframe classification and clustering of content form a basis for meaningful browsing interfaces for lecture videos. While we have demonstrated how automatic approaches for content analysis can be applied to lecture videos, several issues need further exploration. Content matching is based on pixel-comparisons of written material between two images, a process which can only be accurately performed when both images have been captured at the same scale and angle. However, lecture videos are captured in a dynamic environment with more than one camera at widely varying ranges of zoom and tilt. Additional research into vision-based methods of affine or perspective transform parameter determination to normalize images is necessary to better automate and reduce computational complexity of the image matching process. We note that in our previous investigation into edge-based feature matching using the Hausdorff distance, we have found that it is impractical due to its computational complexity.

As we moved to explore unstructured videos, we have omitted visual indices based on classification and content clustering in recent work in order to explore more generic issues of video indexing and browsing that are applicable to a wider range of video genres. Nevertheless, such indices and their visual representations, the Topological View and the Content- and Temporal-based Keyframe browser are novel displays, and informal user feedback suggests that they are most appropriate for lecture videos.
4.2 Textual Indices Derived from ASR

Text cues from ASR and OCR are the most common indices used for multimedia search and retrieval, stemming from user familiarity and efficient descriptiveness. Most research multimedia browsers incorporate some form of text indexing. The Lecture Explorer [Altman et. al., 2002] and Lecture-on-demand [Fujii et. al., 2003] use transcripts for interactive text search queries. The Liberated Learning Project [Bain et. al., 2002] intends to use ASR technology to augment an on-going lecture in real time and provide text transcripts off-line. Some video browsers [Smith and Kanade, 1998] have incorporated filters for transcribed data based on known techniques, such as text frequency - inverse document frequency (TF-IDF). Speech indexing, retrieval, and visualization has enjoyed much attention in domains outside instructional videos, for example SCAN [Whittaker et. al., 1999] for broadcast news stories.

Common to all of these systems is their focus on individual videos, while transcripts and their structural significance across videos have received little attention. We demonstrate approaches for cross-lecture video indexing and referencing for full university courses with 10 to 30 lectures. We take advantage of the relative ease of comparing textual information across lectures, a characteristic that is more difficult when considering visual data (see Chapter 4.1). We first present the methods used in capturing transcripts and discuss the common difficulties encountered in the process. Next, we provide details of the analysis stage and tie in the results with several experimental interactive visualization schemes.
4.2.1 Data Acquisition

4.2.1.1 Transcript Generation

For our purposes, we used course videos from the Columbia Video Network and the commercial Automatic Speech Recognizer IBM ViaVoice to extract transcripts. We have analyzed seven courses from and related to Computer Science with altogether 183 lectures (230 hours of video); four out of these have been analyzed with different instructors’ voice trainings for an additional 90 transcripts. Most transcripts contain between 5,000 and 14,000 words with minimal punctuation marks. Depending on the course structure, a semester of videos comprises between 10 and 30 lectures, where each lecture tends to be 70 or 120 minutes long. Video and audio are highly compressed from the originally videotaped classroom environment to fit between 50 Mb and 110 Mb for effective distribution to distance-learning students; this results in uncomfortably poor reproductive quality.

While the lectures are recorded in a controlled environment with several video cameras and a clip-on wireless microphone worn by the instructor, the levels of technological sophistication and invasiveness on teaching style are rather low. This results in a range of audiovisual quality attributes observed in the compressed videos. The microphone, for example, records not only the instructor’s voice, but also sounds from writing on the board as well as some ambient noise. The audio quality is furthermore impacted by the instructor’s volume level and the position of the microphone with respect to the speaker’s mouth. In summary, while the audio track is just passably good enough for human understanding, it proves to be more problematic to an automatic speech
recognizer. When applying IBM ViaVoice to the extracted audio track, the Word Error Rate is at approximately 75%. We have computed this value by manually transcribing 2 lectures from different instructors and using them as references.

4.2.1.2 Issues of Transcription Accuracy

Glancing over an ASR transcript at first reveals a potpourri of dictionary words, yet a closer comparison to a manual transcript does confirm valid matches of a few (∼25%) distinct phrases. The term “phrase” is used to describe any number of words (≥1) that appear in a semantically meaningful fashion. Table 4.3 exhibits a section from a typical transcription. Besides a modest portion of correctly recognized words, there exist a large number of unique, yet incorrectly identified words (Nafta, assassinations, …). Using known methods of keyword extraction does not establish the desired separation between correctly and incorrectly recognized text. We will later show how undesirable words can be filtered out by using an external corpus of expected index phrases.

Other words unknown to the ASR dictionary may be confused with contextually wrong counterparts, e.g. a “lexer” from “lexical analysis” becomes a “laxer”. Omission of such words may prove problematic in the already limited collection of accurate text, especially if the phrase is a key term. Additional training and dictionary customization may solve this problem.

Training the software with the instructor’s original voice instead of applying some other person’s voice for transcribing a lecture resulted in marginal improvements of only 3% for Word Error Rate. At the same time, the raw number of identified index phrases
Table 4.3. Comparison between manual and automatic transcripts for the course “Analysis of Algorithms”. The Word Error Rate is 71%. Matches are highlighted in bold. Unique, yet incorrect words are marked with *italics*. Words finally used in the indexing tool are CAPITALIZED.

<table>
<thead>
<tr>
<th>Manual Transcript (129 words)</th>
<th>Automatic Transcript (103 words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>… deal with with the data structure like this actually you deal with <em>it with</em> with with heaps also so you have some data <strong>structure</strong> right where where items have names and the <strong>question's</strong> how do you how do you get <strong>how do you get</strong> to the items we've actually you you should have asked this <strong>question</strong> already this semester right uhm so and there this data <strong>structure</strong> doesn't provide a way to find <strong>something</strong> right like a <strong>BINARY TREE provides</strong> if i'm <strong>looking for 27</strong> in a binary tree you know just <strong>given the POINTER to the root of the tree</strong> I have a way to find it right and if you have an <strong>array given you know the name of the array</strong> you have <strong>some way to find</strong> …</td>
<td>… deal with live this <strong>church</strong> is that <strong>CD in do it with</strong> wit of the need all sell <strong>Nafta</strong> this <strong>structure</strong> that will write and <strong>assassinations and the question is how you have to get at it</strong> the added that slate on ye shall ask the **question redhead this vast array of Aum sell and it is its <strong>structure doesn't provide a way to find something</strong> like a <strong>BINARY TREE provides</strong> a way of <strong>looking for 27</strong> and by treat it is <strong>given a POINTER to the router the treehouse where it ought and emulate even though the name Ray Hunt's family finds</strong> …</td>
</tr>
</tbody>
</table>

and their occurrence remained approximately the same at < ±1% (see Table 4.9). However, the qualitative difference between using matched and unmatched speakers was more significant. The difference in uniquely identified index phrases from the same lecture was as much as 20%. The benefits of this substantial difference will be discussed later.

In our experiments, we have not tested language model training, although we recognize that custom dictionaries may improve transcription accuracy. Nevertheless, we categorically rule out training specific to the instructor’s use of language from text
corpora. We can neither use textbook material for prediction of an instructor’s style of language, nor can we request extensive personal documents from each professor for ethical reasons.

While the resulting overall recognition accuracy still remains rather low at 25-30%, we can attribute most of the loss to the poor quality of the recordings. When the five instructors who provided training data used the microphone with a Digital Signal Processing unit at a computer, the Speech Recognizer captured most of the spoken words. These results compare to those from the Liberated Learning Project [Bain et. al., 2002]: With intensive voice training and using special microphones and hardware, the transcription accuracy was 80%. How analysis of casual speech and the creation of custom dictionaries from external sources can lead to improvements in speech recognition of lecture material has been investigated in more detail in [Glass et. al., 2004].

Characteristic of lecture speech is its lack of grammatically accurate sentence structure. This includes repetitions (e.g. “how do you how do you get how do you get”), missing sentence completions (e.g. “…get to the items we’ve <END?> Actually you should have asked this question …”), corrections (e.g. “ … so and there This data structure doesn’t …”), and interjections (“uhm”, “okay”, etc.). While this lack of structure in speech does not map to the careful preparation of a material in a textbook, we are still able to use the external corpus of index phrases to filter out a small portion of key terms from the transcripts. We will also show how an approximate correspondence can be made between lecture transcripts and chapters from the textbook using word pairs.
4.2.2 Analysis

4.2.2.1 Definition of the Target Corpus

For the purpose of indexing, summarization, and cross-referencing, meaningful text needs to be extracted from the transcripts. Ideally, such contents would include “theme” and “topic phrases” that describe the topics covered in a given lecture. We will term them “content phrases”. The term “theme phrase” is loosely defined as a phrase shared among several transcripts, i.e. a phrase that appears in at least ¼ of all transcripts, e.g. “data structure”. A “topic phrase” denotes the opposite, i.e. a phrase shared in less than ¼ of all transcripts, e.g. “binary tree”. The value of ¼ has been experimentally derived from the occurrence patterns displayed in Figure 4.14. Most index phrases are not repeated in more than ¼ of all transcripts, which makes them good candidates for uniquely descriptive phrases. For example, we could expect the lectures of a course in Computer Science Introduction to Data Structures to have common occurrences of the theme phrases “record”, “memory”, “insertion”, and relatively unique occurrences of the topic phrases “push”, “hashtable”, “percolate”. Theme phrases tend to provide a general tenor for the contents of an entire course or a portion thereof, similar to an abstract of a paper or a back cover summary of a book. Topic phrases single out specific topics for one or more lectures, as we would expect from a Table of Contents and chapters of a textbook.

A second category of useful phrases comprises unique “illustration phrases” used in examples and exercises during class lecture. A topic on scheduling algorithms may, for
(A) Course: Database Systems.  
(B) Course: Programming Languages and Translators.  
(C) Course: Analysis of Algorithms.  
(D) Course: Visual Databases.  
(E) Average of 273 transcripts from 11 courses, normalized for number of transcripts.  
(F) Average of 6 textbooks, normalized for number of chapters.  

Figure 4.14. Index Phrase Repetitions in Lecture Transcripts and Textbooks. For a given number of documents (lecture transcripts or textbook chapters on x-axis), each graph returns the number of index phrases (y-axis) that occurred in that number of documents. Most index phrases are not repeated in more than ¼ of all transcripts (vertical line).
example, be illustrated by the pipeline in a “car factory”, and topics in probability and counting tend to be demonstrated with “red”, “green” and “blue marbles”. Including these words in a transcript summary and using them to build an index would be highly desirable. Extracting such phrases is complicated by three observations. Firstly, illustration phrases tend not to be readily available in a standard external index, which would allow us to efficiently find them. Secondly, the low-accuracy transcripts contain a relatively large amount of wrongly recognized unique words, which cannot easily be distinguished from correctly recognized illustration phrases. Lastly, conversational speech in a classroom environment will necessarily include a fair amount of topic-unrelated anecdotal chat between the instructor, the class, and possibly other parties. The difference between meaningful and meaningless contents cannot be easily discerned without additional cues. We have experimentally applied Text Frequency - Inverse Document Frequency (TF-IDF) without significantly successful results; the method captured mostly incorrect phrases, as they outweighed the number of correct ones. A possible solution is to ask the instructor to manually add expected illustration phrases to the standard index used for finding content phrases. In our experiments we have added the illustration phrase “make change” to the index of an “Analysis of Algorithms” course, because it was used for specific examples in dynamic programming. Adding the phrases “java” and “gcc” to the index of a Compiler book proved very effective for the final index phrase visualization as well.
Table 4.4. Selected index phrases from textbook “Introduction to Algorithms” (Cormen, Leiserson, Rivest, and Stein). Phrases have been stemmed and some stop words have been removed.

| amortize analysis | bottom of a stack | avl tree | linear time |
| account method | data structure | binary search tree | matrix |
| aggregate analysis | aa tree | random number generator | problem |
| bob | augmentation | sort | |

4.2.2.2 Filtering Index Phrases

In order to filter out the larger portion of meaningless text from the ASR transcripts, we obtain a corpus of expected phrases and use it as a dictionary of allowable terms. For the purpose of finding an appropriate corpus for lecture transcripts, we employ the course textbook’s index. Since an index generally serves itself as a filter of key phrases for a book, we hypothesize that it can be extended to do the same for lecture transcripts. A large number of phrases found in the index of a textbook are specific enough to fit the curriculum of a course without becoming too generic to fit a lecture in any domain.

The raw index first undergoes some rudimentary word transformations, which will allow for more successful matching to transcripts later on. These transformations are the result of several observations about commonalities between Automatic Speech Recognition, lecture-style speech, and textbook indices. Considerations are made with respect to length of recognized phrases, use of stop words, and grammatical structure. An example of a transformed index is shown in Table 4.4.
Table 4.5. Occurrence of index phrases with different lengths. Using a matched speaker tends to result in slightly more identified index phrases. Unmatched voices, on the other hand, contribute marginally more phrases containing more than one word.

<table>
<thead>
<tr>
<th>Words in Phrase</th>
<th>Matched Speaker (4 Courses, 90 transcripts)</th>
<th>Unmatched Speaker (4 Courses, 90 transcripts)</th>
<th>Matched &amp; Unmatched Speakers (11 Courses, 273 transcripts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23741</td>
<td>23362</td>
<td>59597</td>
</tr>
<tr>
<td></td>
<td>98.15%</td>
<td>98.04%</td>
<td>97.88%</td>
</tr>
<tr>
<td>2</td>
<td>417</td>
<td>432</td>
<td>1208</td>
</tr>
<tr>
<td></td>
<td>1.72%</td>
<td>1.81%</td>
<td>1.98%</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>35</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td>0.12%</td>
<td>0.15%</td>
<td>0.13%</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>0%</td>
<td>0.003%</td>
</tr>
</tbody>
</table>

Given the low-accuracy speech recognition of lectures as well as the casual style of speech, the likelihood of capturing a meaningful phrase decreases dramatically with increasing number of words in the phrase (see Table 4.5). The structure of phrases in a textbook’s index tends to reflect this observation: Most index phrases are 1 and 2 words long when disregarding stop words.

However, not all lines in an index are self-contained entries. Indentations are commonly used in an index to hierarchically mark sub-expressions which are intended to be concatenated with the parent expression (e.g. Table 4.4: “amortized analysis” and “accounting method of” become “accounting method of amortized analysis”). Because of the comparatively low probability of finding the 4-word long index phrase instead of two separate 2-word index phrases, the hierarchical index structure is simply discarded. For the purpose of transforming the index into a dictionary for a set of transcripts, every line of the index becomes one phrase. The reduction of the index to smaller phrases is also performed with respect to stop words in front and after content words, e.g. “accounting
method of” becomes “accounting method”, but “call by value” remains the same. Lastly, a Porter stemmer [Manning and Schütze, 1999] is applied to all words. While a full stemmer truncates many words to their absolute and sometimes unintelligible stems, we apply a partial stemmer that only converts plural nouns to singular nouns, and conjugated verbs to their un-conjugated counterparts. Through experimentation, we have observed that a partial stemmer is in fact more effective for this domain of text analysis.

4.2.2.3 Filtering Word Pairs

As an alternative to finding index phrases in transcripts, we have explored using word pairs. The rationale behind word pairs is to address the relatively incoherent and fragmented order in which contents occurs within a transcript. Since these fragments are padded with stop words and in many cases with repetitions of stop words, we have defined a word pair as two unordered words appearing anywhere within some fixed distance of another. We have empirically determined this distance to be approximately 10 words for the type of transcripts that we are investigating.

The large number of word pairs (at most ten times the number of words in transcript) that is obtained from this analysis is reduced to a smaller set by filtering each word pair by using the textbook index. Only word pairs where both words appear somewhere in the index are relevant. The resulting list of word pairs is on average one order of magnitude larger than the list of index phrases obtained in Chapter 4.2.2.2. From the example in Table 4.6, it is apparent that most word pairs have no coherent semantic meaning, yet some of them do provide some context for the transcript’s contents. While
Table 4.6. Identified word pairs from a “Computer Architecture” course. Some word pairs do not have any semantic meaning, e.g. “multiple very”, yet others are easily recognizable, e.g. “clock cpi”.

<table>
<thead>
<tr>
<th>multiple instruction</th>
<th>multiple processor</th>
<th>million change</th>
<th>call step</th>
</tr>
</thead>
<tbody>
<tr>
<td>multiple operation</td>
<td>million low</td>
<td>call structural</td>
<td>clock instruction</td>
</tr>
<tr>
<td>multiple very</td>
<td>million improvement</td>
<td>call hazard</td>
<td>clock operation</td>
</tr>
<tr>
<td>multiple word</td>
<td>million performance</td>
<td>call instruction</td>
<td>clock cpi</td>
</tr>
<tr>
<td></td>
<td>million time</td>
<td>call compaction</td>
<td>clock per</td>
</tr>
</tbody>
</table>

Table 4.7. Word pairs in decreasing order of log-likelihood based on G² statistics. Almost all of these word pairs have an immediately recognizable semantic significance.

<table>
<thead>
<tr>
<th>register file</th>
<th>operand read</th>
<th>register result</th>
<th>number cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>clock cycle</td>
<td>register cycle</td>
<td>order issue</td>
<td>very simple</td>
</tr>
<tr>
<td>up speed</td>
<td>structure data</td>
<td>cycle instruction</td>
<td>cycle read</td>
</tr>
<tr>
<td>set block</td>
<td>register instruction</td>
<td>history local</td>
<td>little bit</td>
</tr>
<tr>
<td>number block</td>
<td>size block</td>
<td>instruction issue</td>
<td>station reservation</td>
</tr>
</tbody>
</table>

they are not useful for inclusion as key phrases in the user interface, we find that a correlation can be constructed between them and word pairs of a textbook chapter. One of the user interfaces presented later demonstrates how a transcript can be best matched to a chapter in the textbook using word pairs.

Besides using mere occurrence of word pairs, we have also employed the G² log-likelihood statistic to discover significant collocations [Dunning, 1999]. Demonstrated in Table 4.7, the results obtained by using this method are by far more meaningful than word pairs alone. While the log-likelihood statistic is semantically stronger than simple counting, the latter does perform marginally better in establishing correlations between textbook chapters and transcripts, as discussed later.
Table 4.8. Statistics for Index Phrase detection averaged over all lectures in a course.

<table>
<thead>
<tr>
<th>Course</th>
<th>Avg Words in Transcripts /Lecture</th>
<th>Avg Index Phrases /Lecture</th>
<th>Avg Unique Index Phrases /Lecture</th>
<th>Unique Index Phrases /Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matched Spk.</td>
<td>Databases</td>
<td>6121</td>
<td>100</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>Prog. Lang.</td>
<td>7446</td>
<td>249</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>Algo. ’03</td>
<td>7354</td>
<td>414</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Vis. DB</td>
<td>13856</td>
<td>363</td>
<td>50</td>
</tr>
<tr>
<td>Unmatched Spk.</td>
<td>Databases</td>
<td>6182</td>
<td>98</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>Prog. Lang.</td>
<td>7533</td>
<td>258</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td>Algo. ’03</td>
<td>8061</td>
<td>390</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>Vis. DB</td>
<td>14013</td>
<td>373</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Algo. ’00</td>
<td>8038</td>
<td>280</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>Prob. Stat.</td>
<td>5927</td>
<td>30</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Comp. Arch.</td>
<td>7956</td>
<td>159</td>
<td>50</td>
</tr>
</tbody>
</table>

4.2.2.4 Results for Filtering Index Phrases

In performing our analysis on 273 transcripts, we have been able to identify a reasonable number of index terms in the ASR transcripts (see Table 4.8 for details). On average, between 30 and 414 index phrases were found for a given transcript and between 8 and 98 of them were unique occurrences within that transcript. Between 20% and 30% of the index phrases for a transcript had a comparatively significant occurrence between 5 and 50, while between 35% and 50% of them occurred only once. Finally, the number of unique index phrases across an entire course of 10 to 30 lectures was computed to be between 40 and 347 for textbook indices that contained between 253 and 4701 unique index phrases.
Table 4.9. Statistics for Index Phrase detection using the combination of results from matched and unmatched speaker training.

<table>
<thead>
<tr>
<th>Course</th>
<th>Avg Index Phrases /Lecture</th>
<th>Increase over only Matched speaker</th>
<th>Avg Unique Index Phrases /Lecture</th>
<th>Increase over only Matched speaker</th>
<th>Unique Index Phrases /Course</th>
<th>Increase over only Matched speaker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Databases</td>
<td>109</td>
<td>9%</td>
<td>39</td>
<td>18%</td>
<td>136</td>
<td>7%</td>
</tr>
<tr>
<td>Prog Lang</td>
<td>260</td>
<td>4%</td>
<td>69</td>
<td>15%</td>
<td>222</td>
<td>10%</td>
</tr>
<tr>
<td>Algo ‘03</td>
<td>436</td>
<td>5%</td>
<td>116</td>
<td>18%</td>
<td>361</td>
<td>4%</td>
</tr>
<tr>
<td>Vis DB</td>
<td>368</td>
<td>1%</td>
<td>53</td>
<td>6%</td>
<td>106</td>
<td>1%</td>
</tr>
</tbody>
</table>

We observe a qualitative difference between automatically generated transcripts from two different speaker models, one from the course instructor and one from another instructor. We find a significant improvement in detected number of index phrases when both speaker models are applied. Table 4.9 summarizes the improvements for 4 courses; the average number of unique index phrases per lecture increased up to 18%, while the number of unique phrases per course exhibited an increase of up to 10%. The intersection of index phrases from two versions of the same transcript yield mostly rare terms, which are less useful for text indexing. The union of transcripts from two speaker models, however, provides a richer set of index phrases.

The low match rate between transcripts and a textbook index of 5% to 11% can be attributed to a number of external factors that cannot be remedied even with perfect transcriptions. Firstly, university courses do not cover all of the material in accompanying textbooks. Specifically, the courses we have surveyed here cover no more than 50% of the reading material. Secondly, indices contain not only content words, but also names of
individuals and aliases that most of the time are not mentioned in a lecture. Factoring in these observations and realizing that the transcripts are only 25% accurate, the 5-11% figure is not too unrepresentative.

4.2.3 Visualization of Speech Indices

We have investigated several interactive visualization techniques that present the results from text analysis to the student in a meaningful fashion. The three different graphs were developed out of the available dimensions of variability: content from transcripts, content from textbook chapters, identified phrases, occurrence of index phrases in transcripts, and occurrence of index phrases in chapters. Because it is up to the student to decide at what level of detail to view the textual contents (theme versus topic), some of the threshold values were incorporated into the user interface as variable sliders.

Common to all three visualizations are three parameters that are roughly analogous to a camera’s settings. A “zoom” feature allows for setting the specificity of the displayed phrases, ranging from highly topic-specific to entirely thematic. This measure is derived from the occurrence of a phrase across transcripts, where the zoomed-in topic-specific phrase appears in just a few transcripts (1 = lowest), whereas the zoomed-out thematic phrase appears in many transcripts (total transcripts = highest). The “focus” setting denotes the frequency with which a phrase occurs, which is derived from the occurrence of a phrase within a given transcript. The higher the focus is set, the more it is that minimally occurring phrases are removed from display; the display is “sharper”. The third common setting, “contrast”, controls the length of the phrases considered for
display. Increasing this setting bumps out phrases with fewer words, thus creating an emphasis effect on longer phrases.

4.2.3.1 Visualization 1: Transcript Index Map

The Transcript Index Map is a graph in which index phrases are mapped to the transcripts of the lectures they appear in. The purpose of this visualization is two-fold. Primarily it is to provide the equivalent of a textbook index to each transcript, except that the index terms are not ordered alphabetically, but rather in order of occurrence. Transcripts appear temporally increasing along the horizontal direction, and index phrases drop vertically below each transcript in decreasing order of occurrence within a transcript. To further distinguish the frequency with which an index phrase occurs, each item is colored in a spectrum from red to yellow denoting high to low occurrences, respectively. Figure 4.15 shows an index map in which the zoom value has been set to 1, which effectively displays only those terms that appear in no more than one transcript. The result is a collection of topic terms per lecture that describe the contents of that lecture as narrowly as possible, e.g. “aggregate analysis”, “random number generator”, “optimal substructure”, etc.

The second function of the Transcript Index Map is to cross-reference index phrases across consecutive transcripts. For this purpose, semantically equivalent phrases are grouped and their occurrences are summed, effectively increasing their importance in becoming theme phrases. Visually, a grouped item also appears longer, denoting its temporal dependence. An index phrase that appears in five consecutive lectures is
Figure 4.15. Transcript Index Map for the Course “Analysis of Algorithms”: Zoom is set to 1, which displays only those topic phrases which occur uniquely in a given transcript.

Figure 4.16. Transcript Index Map: Zoom is set to 13, i.e. half the number of transcripts for this course. Displayed are topic and theme phrases, with theme phrases appearing in larger blobs, whose width indicates their occurrence across multiple transcripts.
grouped in one entity that now spans those five lectures. As a result, the graph contains differently sized items, which are laid out using a greedy algorithm that fills up as many empty spots as possible nearest to the top. We rationalize this decision by noting that even if the greedy solution is not optimal the relative occurrence of an index phrase is still maintained using color. (In practice, the blobs are usually within five rows of where they should be.) Figure 4.16 shows an index map in which the zoom value has been set to 13, which is half the number of available transcripts. Several theme phrases are now readily available: “graph”, “vertex”, “vertex cover”, “shortest path”, “probability”, etc.

The remaining parameter settings of focus and contrast can be used to further narrow down the displayed index. When increasing the value of focus, less frequently occurring phrases are removed from the graph, thus “cleaning out” terms that may not be as contextually important due to infrequent use. Increasing the value of contrast removes all phrases with less than a certain number of words. The effect of this setting increases the semantic importance of the displayed phrases, because longer phrases tend to carry more meaning, e.g. “binary search tree” versus “tree”.

4.2.3.2 Visualization 2: Textbook Chapter to Transcript Match

In this second visualization we attempt to match a given transcript to a textbook chapter based on the set of identified index phrases. While not every lecture must have a corresponding chapter in the textbook, and while some lectures cover more than one chapter, this interface highlights those chapters that have a relatively high probability of being related to the lecture content. Depending on the actual usage of the textbook by the
Figure 4.17. Chapter Transcript Match for the Course “Analysis of Algorithms”. Lectures are graphed on the vertical axis (increasing downwards) and chapters on the horizontal axis (increasing rightwards). The instructor follows the book in order, which can be seen from the diagonal. The outlier in the rightmost column is additional reading material not covered in the book. Green cells denote correct matches of transcripts to chapters. Yellow cells denote other valid correspondences. Red cells denote incorrect matches.

Figure 4.18. Chapter Transcript Match for the Course “Computer Architecture”, plotted as in Figure 4.17. The instructor focuses mostly on Chapters 2 and 3 of the book and some additional reading material (two right-most columns).
instructor, the display matrix may display a diagonal (see Figure 4.17) if most chapters in the book are covered in order, or the matrix may display a sparse usage of chapters (see Figure 4.18).

The tabular interface is divided into individual chapters from the textbook in columns, and lecture transcripts in rows. Each cell represents a numeric value that ranks the relative score for each chapter-transcript pairing. The score is based on a conceptual three dimensional histogram, whose first dimension is transcript number, second dimension is chapter, and third dimension is phrase identifier (varying from one to total number of phrases in the course). This histogram records for phrase \( k \) the number of times it simultaneously occurs in transcript \( i \) and chapter \( j \), each histogram bin thus is named \( count(i, j, k) \). We define

\[
    score(i, j) = \sum_k \ln(count(i, j, k))
\]

That is, for every phrase in a given transcript \( i \), add the logs of the occurrences of that phrase in chapter \( j \); this approximates a joint probability measure.

We also had studied alternative ways of computing the transcript-chapter match: Instead of using counts of simple phrases, we looked at three different word sets: index phrases, word pairs, and word pairs that had a high \( G^2 \) score (i.e. collocations). Figure 4.19 summarizes the qualitative difference among these three sets, for five courses with altogether 107 lectures, 93 of whose ground truth assignment to one or more chapters in the textbook was obvious. Using index phrases alone, about 50\% of the lectures could be matched to the correct chapter using a zoom value between 6 and 17. Word pairs by
Figure 4.19. Chapter Transcript Matching. Word Pairs, and the combination of Index Phrases and Word Pairs, perform best in matching chapters to transcripts, independent of zoom level.

Themselves achieved around 66% of correct matching in a zoom range between 14 and 26. Using word pairs derived from the $G^2$ measure performed marginally worse at 63%. The combination of index phrases and word pairs resulted in the best average matching rate of 70%. Remarkable is also the robustness at different zoom levels. The range of matching results when disregarding the extreme start and end points is between 61% and 78%.

While textbooks have clear definitions of chapters and sub-chapters, it is unclear what exactly constitutes a “chapter” with respect to this visualization tool. In Figure 4.17 there exists a clear sense of correspondence between chapters and transcripts, while in Figure 4.18, several lectures span one chapter. Individual columns could be split into sub-chapters; however, we found that the accuracy of matching drops about 50%, namely due
Figure 4.20. Multidimensional Scaling of Transcript Similarity based on a selection of index phrases. Lectures on video classification (baseball, documentary, drama, etc.) are clustered near the right, while lectures related to image analysis are closer to the left. In-between is a mixed lecture on both topics. The outlier close to the top is a review session for the entire course.

to the sparsity of book text and therefore sparsity of phrases per sub-chapter (one chapter ≈ 10 sub-chapters).

4.2.3.3 Visualization 3: Lecture Transcript Similarity

For the third visualization of lecture contents for a full course, we have created a graph that visually clusters similar lectures based on a set of selected phrases. The purpose of this tool is to allow a student to explore a course by dynamically grouping lectures that have similar contents based only on a small set of index phrases (see Figure
Closely related transcripts are clustered and linked in red. Weakly related transcripts are linked with a color that fades into the background, while unrelated transcripts are not linked at all. Multidimensional Scaling (MDS) is used to collapse the higher dimensional space of \( N \) lecture transcripts to two dimensions. The distance matrix used for MDS is constructed by means of the Dice Distance \([\text{Dice}, 1945]\) applied to each pair \((i,j)\) of all transcripts:

\[
\text{dist}(i, j) = \frac{b + c}{2a + b + c}
\]

where \(a\), \(b\), and \(c\) are the co-occurrences of all phrases (a) in transcript \(i\) and \(j\), (b) not in transcript \(i\) but in transcript \(j\), and (c) in transcript \(j\) but not in transcript \(i\).

We have found that semantically meaningful contents, such as index phrases, produce distinguishable graphs. Closely related lectures appear in clusters, while largely unrelated lectures produce outlier nodes. Figure 4.20 shows a graph for the selection of phrases “baseball”, “classification”, “documentary”, “drama”, “home video”, “musical”, “newscast”, “sitcom”, “soccer”, and “video” from a course in “Visual Databases”, which covers topics on image and video analysis, retrieval, summarization, and visualization. These video classification terms appear mostly in lectures 6, 8, 9, 10, 11, and 12, which can be seen clustered on the right of the graph (Note: 9 and 10 overlap). Lectures 3, 4, and 5 cover image retrieval and face recognition and thus appear farthest away near the left of the graph. Centered between these two clusters we find lecture 7, which discusses jpeg and mpeg algorithms; this also corresponds to a “semantic average” between images and video. An outlier in this visualization is lecture 13 near the top; it serves as a review session of the entire course.
4.2.4 Conclusions

Textual index browsers presented here have been explored specifically for sequences of lecture videos. While they can be adapted to videos of other genres, the Transcript Index Map is the only universally applicable interface. For any sequence of videos, it visually emphasizes terms and phrases used across videos, thereby creating key themes shared by those videos. The Chapter Transcript Map lends itself particularly well to a series of videos, whose content maps into a comparable collection of texts, e.g. lecture videos mapping to chapters in a textbook. Videos in other genres, such as presentation, meeting, or interaction videos typically do not exhibit such a mapping, because they represent temporally discrete events, whose content does not follow an extensive pre-written text corpus. Finally, we do not further explore the Multidimensional Scaling-based Transcript Similarity interface, but we note its potentially useful application to clustering video search results. Similar to the approach of clustering lecture transcripts on selected phrases (see Figure 4.20), videos from text-based search results can be clustered to suggest similarities.
4.3 Task-Based Experiments and Evaluation

We have evaluated our visual approaches of lecture video browsing in a user study with 11 students and one professor. We use an earlier version of the lecture browser interface, which features a keyframe browser, topological view, and information mural (see Figure 4.21). The keyframe browser features thumbnails of keyframes, clustered by similar content and displayed in vertical film strips. When the mouse cursor is moved over a thumbnail, a semi-transparent magnifying glass with full-sized keyframes is overlaid in the interface. The topological view displays the same clusters as abstract icons with indications of temporal recurrence. The information mural is a similar display, which emphasizes temporal duration of clustered keyframes. Visual elements in all three frames are selectable and are inter-linked.

The experiment was set up to measure user interaction with these user interface elements for a variety of tasks (see Table 4.10) typical of content retrieval from lecture videos. Among many observations as a result of the user study, we find that the keyframe browser and the topological view are the most useful UI elements.

In the user study, each participant is presented with two sets of 11 questions, each for one of two lecture videos from the course “Advanced Database Systems”. We randomize the order of the two videos to more accurately determine the presence of a learning curve between the two sets. A task is completed successfully by selecting a correct keyframe; for each task, one or more keyframes have been identified a priori as correct answers. In our logs, we keep track of the number of incorrect attempts before a task is correctly completed.
Figure 4.21. Early Lecture Video Browser used for User Study. Compare with Figure 4.1 and Figure 4.2. The interface features a keyframe viewer with semi-transparent magnifying glass for high-quality keyframes, an information mural and a topological view for abstract visualizations of clustered contents. A keyframe player can be used to simulate fast video playback, while an external media player is provided to playback full video content. In the final version of this interface, the information mural was dropped, because it was not well received by users. The topological view and keyframe viewer were fused into one interface, and semi-transparent pop-ups were moved outside of the keyframe viewer, because they distracted users from the underlying keyframe structure UI. The User Study frame displays questions and gives feedback of task progress.
Table 4.10. User study tasks for lecture video browser evaluation.

<table>
<thead>
<tr>
<th>Task</th>
<th>Example formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locate a specific keyframe given a visual example</td>
<td>“Locate this keyframe”</td>
</tr>
<tr>
<td>Locate a keyframe of a media type</td>
<td>“Locate a slide that displays contents from the computer or … shows the professor actively addressing the class”</td>
</tr>
<tr>
<td>Locate a topic (teaching unit)</td>
<td>“Locate a slide from the discussion on …”</td>
</tr>
<tr>
<td>Locate the main topic</td>
<td>“Locate a slide from the main topic for this class”</td>
</tr>
<tr>
<td>Locate the beginning of a topic</td>
<td>“Locate the slide that begins the discussion on …”</td>
</tr>
<tr>
<td>Locating the end of a topic</td>
<td>“Locate the slide that ends the discussion on …”</td>
</tr>
<tr>
<td>Locating the most interaction</td>
<td>“Locate a slide in the portion of the class where the professor switches the most between the different topics on the blackboard”</td>
</tr>
</tbody>
</table>

Table 4.11 summarizes average time required for task completion and the average number of incorrect selections per completed task. We find the highest error rates among tasks related to finding structural information, e.g. locating the beginning or end of a topic, which may be due to misinterpretation or shortcomings of the interface in communicating this information. Table 4.11 also shows the difference in time and rate of incorrect responses between the first and second set of questions. In general, we find that tasks related to structural content are completed faster and with fewer incorrect attempts during the second period of the user study. Participants quickly gain an understanding of the interface and how it is best used for various tasks.
Table 4.11. Statistics for user study tasks. Learning curve is measured as the change of time required and rate of incorrect responses between two sets of user study tasks.

<table>
<thead>
<tr>
<th>Type of Task</th>
<th>Average Time (sec)</th>
<th>Average # Incorrect</th>
<th>∆ Avg. Time (sec)</th>
<th>∆ Avg. # Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locate a specific keyframe given a visual example</td>
<td>69.15</td>
<td>0.08</td>
<td>+28.06</td>
<td>-0.17</td>
</tr>
<tr>
<td>Locate a keyframe of a media type</td>
<td>23.61</td>
<td>0.42</td>
<td>-3.91</td>
<td>+0.17</td>
</tr>
<tr>
<td>Locate a topic (teaching unit)</td>
<td>115.52</td>
<td>0.95</td>
<td>-10.34</td>
<td>+0.24</td>
</tr>
<tr>
<td>Locate the main topic</td>
<td>59.78</td>
<td>0.33</td>
<td>-11.65</td>
<td>0</td>
</tr>
<tr>
<td>Locate the beginning of a topic</td>
<td>89.22</td>
<td>0.92</td>
<td>+11.61</td>
<td>-0.7</td>
</tr>
<tr>
<td>Locating the end of a topic</td>
<td>54.46</td>
<td>1.08</td>
<td>-42.56</td>
<td>-0.83</td>
</tr>
<tr>
<td>Locating the most interaction</td>
<td>99.54</td>
<td>1.7</td>
<td>-47.77</td>
<td>-0.28</td>
</tr>
</tbody>
</table>

Table 4.12. Usage of various features: average time in seconds spent in each visualization frame.

<table>
<thead>
<tr>
<th>Set</th>
<th>Slide Frame</th>
<th>Topological View</th>
<th>Information Mural</th>
<th>Slide Player</th>
<th>User Study Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>855.3</td>
<td>67.4</td>
<td>2.7</td>
<td>5.3</td>
<td>40.7</td>
</tr>
<tr>
<td>2</td>
<td>762.2</td>
<td>68.1</td>
<td>2.8</td>
<td>15.8</td>
<td>31.3</td>
</tr>
<tr>
<td>1 and 2</td>
<td>1617.5</td>
<td>135.5</td>
<td>5.5</td>
<td>21.1</td>
<td>72.0</td>
</tr>
</tbody>
</table>

To better evaluate interface usage, we measure time spent in each visualization frame and determine transition patterns between them. While time spent in a particular UI establishes its popularity, transition patterns identify which visualization frames would benefit from better fusion of functionality. Table 4.12 illustrates the usage of UI elements in the lecture browser during the user studies. The Slide Frame is used most (87% of the
time), which is likely motivated by its rich content of thumbnails and its interactive magnifying glass. The Topological View is used much less (7%), but nonetheless significantly enough to be considered a valuable tool. Without counting the necessary use of the User Study Frame used to display tasks and provide feedback during a user study, the Information Mural and Slide Player were not used with any notable frequency.

Transition patterns between UI frames are determined by pairs of consecutive actions, including two successive actions within and between UIs. The transition matrix (see Table 4.13) illustrates that most of the activity is confined to the Slide Frame, which is due to the high degree of interactivity with thumbnails and the magnifying glass. We note that the only other frequent transitions take place between the Slide Frame and each of Topological View, Slide Player, and User Study Frame. Transitions between Slide Frame and User Study Frame are not contextually meaningful, because we expect participants to complete all tasks by selecting a keyframe, which is necessarily followed by a user study action.

Using the transition matrix, we illustrate the user interaction with various UI elements in a transition diagram (see Figure 4.22), in which size of circles and thickness of edges has been scaled to represent the intensity of interactions. Circle area (with diameter \( d \)) is scaled as a function of time in seconds spent in the corresponding UI frame:

\[
d = 2 \sqrt{\frac{time}{\pi}}.
\]
Table 4.13. Matrix for intra- and inter-transitions of visualization frames. Values denote average number of actions.

<table>
<thead>
<tr>
<th>To</th>
<th>Slide Frame</th>
<th>Topological View</th>
<th>Information Mural</th>
<th>Slide Player</th>
<th>User Study Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slide Frame</td>
<td>1672.2</td>
<td>14.7</td>
<td>1.0</td>
<td>13.5</td>
<td>37.4</td>
</tr>
<tr>
<td>Topo. View</td>
<td>18.8</td>
<td>15.8</td>
<td>0</td>
<td>1.2</td>
<td>0.5</td>
</tr>
<tr>
<td>Info. Mural</td>
<td>1.0</td>
<td>0</td>
<td>0.3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Slide Player</td>
<td>12.9</td>
<td>1.6</td>
<td>0</td>
<td>0</td>
<td>0.2</td>
</tr>
<tr>
<td>User Study F.</td>
<td>25.3</td>
<td>1.5</td>
<td>0</td>
<td>0</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Figure 4.22. Diagram for Intra- and Inter-Transitions in Lecture Video Browser. Size of circles and thickness of arcs are mathematically scaled to represent the frequency of transitions. Insignificant values are omitted.

Arc thickness (width $w$) is scaled as a function of number of transitions between two UI frames. Here, we scale the numerical values logarithmically to account for the large differences, and then linearly scale-up the resulting values:

$$w = 3 \times \log(\text{actions})$$
For lack of significance, the Information Mural and rare transitions are omitted from the graph. Without considering the User Study Frame for content retrieval, we observe that the most inter-frame interactions occur between the Slide Frame and the Topological View, and between the Slide Frame and the Slide Player. Because these user interfaces are visually disjoint in the lecture browser, but are heavily used in combination, we speculate that better spatial juxtaposition of UI elements that are used together would lead to improved browsing. An updated interface is illustrated in Figure 4.23.

Surveys collected from user study participants confirm that the most useful features for lecture video content retrieval include the Slide Frame and the Topological View, and the least useful is the Information Mural. The Slide Frame’s organization of keyframes into clustered units of similar content and its browsing features were deemed
most useful. However, the majority of users agreed that the semi-transparent superimposition of full-sized magnified keyframes over the thumbnail interface, while an interesting feature, was distracting. Most participants would prefer high-quality keyframes displayed outside of the area used for mouse scrolling.
4.4 Course-Study-Based Experiments and Evaluation

We have made available the VAST MM browser, after three years of improvements, to students from two core computer science courses in the Fall 2007 semester for their final exam preparation to measure a potential impact of the tool on course study. The actual browser UI is shown in Figure 3.2 to Figure 3.4, one of which reappears as Figure 4.24. The courses include “Computer Architecture” and “Programming Languages and Translators” with a distribution of undergraduate and graduate students, most of whom major in Computer Science. Lecture videos are typically available only to long-distance students, and therefore, participants would not have had access to the videos otherwise. We introduced the tool one to two weeks before the final exam, so that students had access to the resource during their reading period. Limited instructions of the tool’s features were given in a short in-class demonstration. While all students had access to the tool, participation was entirely voluntary, and of the 142 students from both courses, 91 (64%) did not attempt using it at all.

We compute a normalized difference in midterm exam scores and final exam scores to create a measure of improvement. In this aggregation, we include all students, regardless of their participation. The normalized measure is computed in terms of difference of standard deviation from the mean for each exam:
Figure 4.24. VAST MM Video Browser used during Final Exam Preparation.

$$\mu_{\text{midterm}}$$  Mean of midterm exam grades, all students

$$\sigma_{\text{midterm}}$$  Standard deviation of midterm exam grades

$$\alpha_i$$  Midterm exam grade for student $i$

$$\mu_{\text{final}}$$  Mean of final exam grades, all students

$$\sigma_{\text{final}}$$  Standard deviation of final exam grades

$$\beta_i$$  Final exam grade for student $i$

$$\Delta_{\text{improvement}} = \frac{\beta_i - \mu_{\text{final}}}{\sigma_{\text{final}}} - \frac{\alpha_i - \mu_{\text{midterm}}}{\sigma_{\text{midterm}}}$$
Using Δimprovement values we are able to test statistically whether the use of the VAST MM browser had any significant effect on the improvement of student’s performance.

A note on statistical significance: Parametric model structures are specified a priori, whereas non-parametric model structures are determined based on the data. The data obtained for the improvement in students’ grades are not considered to be a parametric measure. Therefore, we must apply a non-parametric model, the Wilcoxon signed-rank test, to test the significance of grade improvement based on the usage of VAST MM. Similar to its parametric form, the paired t-test, the Wilcoxon tests the hypothesis:

\[ H_0 : \mu_{\text{non-users}} = \mu_{\text{users}} \]
\[ H_a : \mu_{\text{non-users}} < \mu_{\text{users}} \]

Our null hypothesis states that VAST MM users achieve the same exam average as non-VAST MM users, while \( H_a \) states that non-VAST MM users achieve lower scores. We conduct the test at \( \alpha = 0.05 \) and obtain a \( p \)-value < 0.0001, allowing us to reject our null hypothesis, \( H_0 \).

Figure 4.25 presents distributions of exam difference for the two groups of participants. On average, non-VAST MM users experience a slight drop of 0.07, whereas VAST MM users improve by 0.29 standard deviations. The data presented here is representative only of students who have used the tool for more than 30 minutes.
Figure 4.25. Midterm-to-Final Exam Grade Improvements for students who have used the VAST MM resource (top graph) and students who have not (bottom graph). Average improvement is 0.29 standard deviations in the presence of VAST MM and -0.07 in its absence. The absolute difference is one third of a standard deviation.

We determine that duration of usage is a strong indicator for improvement. At $\alpha = 0.05$ and using a two-tailed Pearson’s correlation, we observe a significant positive trend between duration of VAST MM use and improvement in grade: $r = 0.169 \ (N = 25)$ for students who used the tool more than 30 minutes, and $r = 0.301 \ (N = 51)$ for all participants. Figure 4.26 graphically summarizes the time students spent using VAST MM versus their improvement in exam grades.

We apply another Pearson's correlation test to determine VAST MM browser features correlated to grade improvement. We observe strong correlations at $\alpha = 0.05$ for the following features:

- Skimming of keyframes
- Zoom activity
- Scene segmentation activity
Text context activity (changing clusters of keywords and phrases)

Viewing actual video

Selecting videos from library

Low segmentation value (i.e. larger number of thumbnails displayed for skimming)

It is generally difficult to measure the isolated effect of one variable in the presence of many unaccounted ones. As such, the demonstrated improvement of exam scores for students using the VAST MM tool could be attributed to other effects, such as extensive study from textbooks and notes. However, because we compare two exam study periods in our evaluation, one with and one without the availability of video resources through VAST MM, we can eliminate bias caused by several external factors.
Under the assumption that students did not alter their fundamental study patterns between midterm and final exams, we can conclude that the use of VAST MM for video reviews was beneficial.
Chapter 5

Presentation Videos

With the growing use of videos in classrooms other than for recording lectures, we investigate a novel application of video libraries for classroom presentation videos. A presentation video contains one or more distinct presentations carried out by students or teams of students. Typically, an electronic medium like PowerPoint is used to accompany the speaker; however, for the purpose of segmentation and visualization, it is not required here. The recorded videos differ from lecture videos in several critical respects: their shots are longer without distinct visual cuts, presentations are carried out by multiple speakers, audio quality varies significantly, and a repetitive formal structure of keywords exists across presentations. Intrinsic but unpredictable aspects of these video records are the inclusion of short video sessions, speaker transitions, question and answer sessions (of varying audio quality), interruptions, cameraperson errors, etc. (see Figure 5.1).
Figure 5.1. Examples of Imagery from Typical Presentation Video. Segmentation of such videos should take advantage of cues from audio, video, and text, leading to integrated approaches for summarization and indexing.

Figure 5.2. Split Panoramic View of Presentation Classroom. Presentation videos capture presentations in the style of the middle and right views at various zoom levels. During Q&A sessions, the camera sometimes pans to the audience.

A single camera captures the speaker standing next to the projected image (see Figure 5.2), and a handheld or stationary wireless microphone is used by the speaker to better pick up the audio signal. The video is neither shot professionally, nor is the presentation space set up specifically for the purpose of video productions. It is not uncommon for students to forget to speak into the microphone, or to appear in the filmed
frame. The time period during which a speaker presents does not necessarily overlap with the projected slides; two presenters may share a slide. Separating segmentation and visualization of audio and video is especially useful for these conditions. Although good practice would suggest that the video could be easily segmented by speaker, these and other frequent violations of presentation rules make a more robust approach necessary.

The methods and tools discussed here address the needs of two audiences involved in the presentations: instructors and students. Presentations are used by instructors to evaluate and grade the performance of teams and individual students. Recorded video material is used to revisit the presentations as necessary in some cases to re-evaluate and discuss with students. However, the inherently serial nature of video inhibits the instructor from quickly locating portions of the presentation.

With the introduction of video-taped presentations for archiving purposes, students have gradually become more interested in reviewing this material as well. Since presentations are recorded at the midterm and at the end of the semester, students have found it useful to evaluate their own and their peers’ performance towards improving presentation skills. The corpus of presentation videos considered here originates in a collection of videos over three years. Since video has become a popular medium of self-evaluation, especially in the educational environment, we foresee the continued production and persisting problems of indexing, browsing, searching, and disseminating such presentation videos.
5.1 Visual Indices

Unedited presentation videos do not feature production cues, such as scene cuts or fades, which are usually used to define visual segmentation. Unlike professional presentations, student presentations are held in classrooms, where presenters and audience inevitably appear in the same camera shots. Cues for visual segmentation of such noisy video data should be sensitive to the recording environment. It is neither sufficient to rely on only one cue, nor is it appropriate to pre-define the characteristics of a shot. We therefore apply methods that generate a visual segmentation, and in the user interface allow the user to select the granularity.

5.1.1 Visual Feature Extraction

When presentations make use of electronic slides and the camera captures their content, slide changes are used as a cue to indicate an interesting visual change. Abrupt visual activity of this kind is similar to scene cuts in edited video. We use a windowed approach that detects significant visual change $V_k$ in the immediate neighborhood ($\pm$ two seconds) of a point $p_k$ in the video.

We first define a low-level measure for frame-by-frame differences. We compute consecutive frame differences using pixel intensity change between two frames. A unit of intensity change between two pixels is counted when their intensity difference is greater than 30 (out of 255). We impose this binary rule to reduce the effect of very high intensity change on a pixel-level, because we are primarily interested in whether a
significant change occurred rather than its intensity. To further attenuate the effect of sporadic changes in small regions of the video, we divide a video frame (e.g. 640x480, or 320x240) into a 10x10 grid and impose a limit of 10% of cumulative intensity change that each sub-region contributes to the global measure. That is, if more than 10% of a sub-region’s pixels exhibit significant visual change, only pixels up to that limit are counted towards the frame pair’s overall measure. The raw measure of visual change between two consecutive frames is computed as the sum of visual changes in the 10x10 grid; it has a non-normalized raw numerical range of [0..10].

A scene cut based on abrupt visual change is determined between two video frames (point $p_k$) in a sequence of frames when their low-level abrupt visual change measure deviates enough from that of adjacent frames (± two seconds). This mid-level measure (see Figure 5.3) is designed to isolate spikes in a series of relative calm without relying on absolute thresholds. Using it, we are able to detect abrupt scene changes in variably noisy visual settings.
Our second method of detecting interesting visual events is based on gradually changing content in the video, such as camera pans, zooms, entering and/or leaving of a person with respect to the camera view. We use a windowed approach that compares histogram differences between more distant frames (four seconds apart).

We first define a low-level measure for distant frame pair differences (see Figure 5.4). We compute distant frame differences using color histogram change between two frames. Experimentally, we have determined that a color histogram with two bits per color (red, green, blue) resulting in 64 bins is sufficient for capturing this information. The sum over bin differences between two frames at most equals to twice the number of pixels in a video frames, and therefore, after dividing by the frame dimensions, this measure has a numerical range of $[0..2]$.

A scene cut based on gradual visual change is determined at a point $p_{k - 2 \text{ sec}}$ in the
Figure 5.5. Mid-Level Gradual Visual Change Detection based on the low-level gradual visual change measure.

where $V_k(p_{k-2\text{sec}}) \equiv A \land B \land (C \lor D)$, where

- $A = \max(p_{k,\text{left}}) < \max(p_{k,\text{right}})$
- $B = \max(p_{k,\text{left}}) < \text{mean}(p_{k,\text{center}})$
- $C = \max(p_{k+1,\text{left}}) \geq \max(p_{k+1,\text{right}})$
- $D = \max(p_{k+1,\text{left}}) \geq \text{mean}(p_{k+1,\text{center}})$

video when histogram color changes are detected over an extended temporal period (about two seconds) with relative significance. Specifically, this mid-level measure (see Figure 5.5) tests for an increase in histogram color change over distant (four seconds) frames (Figure 5.5, test $A$), it ensures that this increased activity persists throughout the temporal distance of 4 seconds and is not a spurious occurrence (Figure 5.5, test $B$). It also tests for two conditions that signal the end of gradual color change, only one of which must be met: a test for a sharp decrease (Figure 5.5, test $C$), or a test for a slow decrease (Figure 5.5, test $D$).

In a final, high-level step, visual activity from both methods is combined to generate an aggregate set of visual events. Using a windowed approach, gradual and abrupt changes are merged. If activity from both is recorded in close vicinity (e.g. 4 seconds), the temporal location of abrupt change is preferred. Both measures of visual change are included as tunable parameter in the user interface, termed “Scene Segmentation”.
5.1.2 Visual Feature Control

Visual cues are extracted from videos in a pre-determined fine granularity, taking into consideration visual change. For the eventual representation in the browser user interface, the measure used during segmentation is too fine and would result in more than desired redundant keyframes showing visual change. The numerical measures representing visual change also depend largely on video-intrinsic parameters – some videos exhibit more subtle characteristics of visual change than others. For example, frame-by-frame differences are significantly influenced by the camera’s zoom parameter.

In general, for unstructured videos, we find that there exists no universal threshold that optimally decimates the amount of visual information to a desired set of visual cues. It is equally as infeasible to predict a comfortable level of visual segmentation and amount of visually representative cues for all users. This parameter is to the most part dependent on individual user preference. We have therefore moved threshold determination into the user interface as a freely adjustable parameter. It determines the level of visual segmentation granularity and implicitly adjusts the number of keyframes displayed in the video summary. At one extreme, this parameter causes keyframes displayed in almost constant short intervals (see Figure 5.6) and at the other extreme, it sets the number of keyframes to a minimum of only a few per video (see Figure 5.7).

A second freely adjustable user interface parameter, the temporal zoom slider, controls the duration of video cues displayed. The parameter adjusts the amount of temporal information per pixel in the user interface, ranging from 10 to 0.1 seconds per
Figure 5.6. Video Summary with Parameter "Scene Segmentation" set to fine granularity (low visual change threshold), causing it to display a large amount of keyframes.

Figure 5.7. Parameter "Scene Segmentation" is set to coarse granularity. Keyframes are displayed for scenes with significant visual boundaries.

Figure 5.8. Video Summary with Parameter "Zoom" set to a value which displays more temporal information per pixel. In this case, the screen displays 60 minutes of video summary data.

Figure 5.9. Parameter "Zoom" is set to a value which displays less information per pixel. In this case, 90 seconds of video information are displayed on the screen.

pixel. At the one extreme, the parameter causes the interface to display a large amount of information in a relatively small amount of screen area (see Figure 5.8), useful for overview browsing, and at the other extreme, the parameter stretches the timeline to display a short amount of time and hence less information with the intent of precise browsing (see Figure 5.9). In combination with the visual segmentation slider, the user
can adjust two dimensions of information visualization – temporal precision and temporal density.

5.2 Audio Indices

Audio data provides a rich set of cues for video indexing. In addition to text-from-speech, audio lends itself to determine when actors are speaking, for how long they are speaking, when they recur, and who they are interacting with in a dialog. The audio track also contains information that is otherwise difficult or impossible to extract from pure visual information, such as emphasis and speech patterns inherent to a person or indicative of certain interesting characteristics.

We explore speaker segmentation and clustering intended for summarization of speaker events in presentation videos. We demonstrate shortcomings in popular speaker clustering metrics and present a correction based on our observations. Finally, we introduce a visual face index based on speaker segmentation cues for video browsing.

5.2.1 Speaker Segmentation

In general, audio segmentation for presentation videos lends itself to segmentation by speaker. We use the method of detecting speaker changes via the Bayesian Information Criterion (BIC) [Chen and Gopalakrishnan, 1998]. The BIC is applied as a measure for maximum likelihood of detecting a change in the audio stream, which is modeled as Mel Frequency Cepstral Coefficients (MFCCs). Cepstral coefficients are used to represent speech signals in a low-dimensional space by modeling the rate of change in
various spectrum bands. Mel frequency refers to the mel scale of spectrum bands, that is a logarithmic scale which better reflects the human auditory system than a linear scale.

A single change in speaker is detected at point $i$ in a short audio segment $S$, represented as $N$ 13-dimensional MFCC vectors, by computing a ratio statistic ($R(i)$) for sub-segments $A=S_{i\ldots j}$ and $B=S_{(i+1)\ldots N}$:

$$R(i) = N \ast \log \left| \Sigma_a \right| - i \ast \log \left| \Sigma_{ai} \right| - (N - i) \ast \log \left| \Sigma_{a1} \right|,$$

where $\Sigma_A$ is the covariance matrix of MFCCs in sub-segment $A$, $\Sigma_B$ the covariance matrix of MFCCs in sub-segment $B$, and $\Sigma$ the covariance matrix of MFCCs in both sub-segments combined.

The $BIC(i)$ is then defined as:

$$BIC(i) = R(i) - \left[ \frac{1}{2}(d + \frac{1}{2}d(d + 1)) \ast \log(N) \right].$$

If $BIC(i)$ is positive, a significant enough difference between the Gaussian distributions for sub-segments $A$ and $B$ has been found, therefore signaling a speaker change. In practice, this calculation is impractical to perform at every point $i$ in the audio stream. We therefore first apply a coarse segmentation at every second, and, if a speaker change is determined, we apply a fine segmentation at every 100 milliseconds to localize the change with more precision.

Multiple speaker changes in a long audio stream are detected by iteratively measuring the length of the $A$ audio segment, and when a speaker change is found, removing segment $A$ (i.e., FIFO). The process is outlined in Figure 5.10. Figure 5.11
Figure 5.10. Speaker Segmentation via BIC

Figure 5.11. Speaker Change Detection via BIC. Circles mark the points in time at which a speaker change occurs. A speaker change is detected when a maximum BIC value above 0 is measured.
gives an example of an eight minute audio stream, in which 10 speaker changes have been identified. The maxima between speakers’ BIC values can be clearly located.

We have evaluated this method on presentation videos, and we have found that the best segmentation is achieved when sub-sampling the audio track as follows: For every 1/8th second (roughly one syllable), we compute one MFCC vector of length ≈ 1/64th second. The resulting audio segment length is roughly equivalent to 128 samples (in 12 kHz sampled audio), 256 samples (in 16 kHz sampled audio), or 512 samples (in 32 kHz and 40 kHz sampled audio), a power of two size necessary for FFT. We have experimented with several other ranges of values for sampling frequency (8kHz, 16kHz, 32kHz), Fast Fourier sampling windows (256, 512, 1024), and MFCC vectors per second (256 to 125). However, selecting too many or too few MFCC vectors per second, or selecting much longer or shorter sample sets, results in dramatic over- or under-segmentation.

The results from speaker segmentation are very favorable. In our experiments we observe less than 10% false negatives and positives. False positives tend to be introduced by the occurrence of small pauses in the audio track (that is, silence is detected as a different “speaker”). The final segmentation, as well as the raw audio activity graph (audio amplitude graph) are included in the user interface (see Figure 3.2, Figure 3.3, and Figure 3.4).
5.2.2 Speaker Clustering

The KL2 distance (see Equation 1) is used frequently in the context of speaker clustering based on MFCC features, but it has some problems, particularly for short segments or segments of greatly unequal lengths. It is not always clear from related literature under what conditions clustering took place. For example, in [Siegler et. al., 1997] speaker data from news broadcasts is clustered using KL2, but it is not clear whether speech segments are equal or highly similar in size; in the news domain, however, speech segments tend to be short, given frequent cuts to other anchors and reporters. Secondly, the authors of [Huang and Hansen, 2004] argue that the Bayesian Inference Criterion (BIC) and other distance metric based methods often suffer in estimation error due to insufficient data, i.e. short segments. They observed degraded performance for segments of very short duration, e.g. two seconds, and attribute this phenomenon to insufficient data in the estimation of the covariance. Thirdly, work in [Lu and Zhang, 2002] presents a modified KL2 distance measure, in which the mean component (i.e. $M(A,B)$ in Equation 3) is removed. The authors argue that the mean is easily biased due to various environment conditions. Most (80%) of their speaker segments are between 3 and 15 seconds in length. They also have empirically observed that very short segments (< 2-3 seconds in length) dramatically decrease performance of segmentation and tracking. Lastly, work in [Li and Dorai, 2004] uses the KL2 distance for clustering speaker segments in the domain of instructional video. It is unclear what size distribution their speech data has, although we can assume significant differences in segment lengths given the genre of video.
5.2.2.1 KL2 Clustering Approach

The symmetric KL distance is defined for two given random variables A and B with Gaussian distributions as:

\[ KL2(A, B) = C(A, B) + M(A, B) \]

Equation 1 Symmetric KL distance (KL2)

\[ C(A, B) = \frac{1}{2} \left[ tr(\sigma_A^{-1} \sigma_B) + tr(\sigma_B^{-1} \sigma_A) \right] - d \]

Equation 2 KL2 Covariance term

\[ M(A, B) = (\mu_A - \mu_B)(\sigma_A^{-1} + \sigma_B^{-1})(\mu_A - \mu_B)^T \]

Equation 3 KL2 Mean term

where \( \sigma \) is the covariance matrix, \( \mu \) the mean vector, and \( d \) is the dimensionality of the feature vector (here, the MFCC vector of length 13). A typical value for dimensionality \( d \) is 13 for MFCC features, representing the energy coefficient \( d(0) \), and the 12 first MFCC coefficients \( d(1) - d(12) \).

Based directly on its derivation in [Campbell, 1997], the KL2 function is comprised of two parts: a portion, \( C(A,B) \), which is strictly computed from the covariance and a portion, \( M(A,B) \), which also includes the mean vectors of the feature set. \( M(A,B) \) often directly reflects, among other physical influences, possible environmental changes in the audio source, e.g., volume shifts. Some work suggests that results improve when this term is not considered at all [Li and Dorai, 2004].

5.2.2.2 KL2 Simulation

Since KL2 is a distance measure, feature sets with similarities produce a KL2 value towards zero, while segments with differences produce KL2 values > 0. We observe significant differences in the KL2 measure for feature sets from one speaker in which at least one segment is short. And, during clustering, long audio segments (> 20
seconds) also are observed to converge to clusters faster and with more accuracy than shorter segments. Therefore, we set up two simulations to test the KL2 measure for these observed length effects. In Experiment 1, random datasets of variable-sized MFCC feature sets are generated, and their simulated KL2 distances are determined in a matrix of $(\text{segment A length}) \times (\text{segment B length})$. In Experiment 2, MFCC features from a single speaker are used to create new variable-sized feature sets, and their KL2 distances are computed in a similar matrix. The first experiment uses simulated data to illustrate the general trend in KL2 over differently sized speech segments (see Figure 5.12), while the second experiment validates this trend on real data (see Figure 5.13).

5.2.2.3 Observations

The KL2 measure for comparisons between differently sized speech segments shows significantly degraded results depending on the length of the shorter segment. Additionally, smaller feature sets have obvious disadvantages over larger ones. A closer analysis of the KL2 metric, which outlines the source of the length effect, is presented in Figure 5.16 and Figure 5.17. The second term of $C(A,B)$, itself an asymmetric distance $tr(\sigma_B^{-1} \sigma_A)$ for speech segments A and B, responds to the length of B strongly in what appears to be an inverse power fashion (see Figure 5.16); variations due to the length of A are more moderate (see Figure 5.17). The symmetric KL2 distance therefore includes two such asymmetric terms, which behave similarly to each other (see Figure 5.15).

The reason behind the degraded KL2 distances for short speech segments is the lack of sufficient samples that can be used to accurately model the parameters of the
Figure 5.12. KL2 Distance Length Effect. Simulation on randomly generated normal distributions: Comparing two small-sized feature vector sets (< 200 vectors) introduces estimation errors due to limited sampling size.

Figure 5.13. Real Speech Data: random subsets (1-252 seconds) from one speaker’s feature set, compared pair-wise. Comparisons of short segments (one or both) result in higher KL2 distances and thus less similarity. Note that the vertical axis is 4 times that of Figure 5.12.

Figure 5.14. KL2’ Responses for Real Speech Data, compensating for length effect. The response of KL2’ more uniformly reflects distance. Note that the vertical axis is four times that of Figure 5.12.

Figure 5.15. Term $\text{tr}(\sigma_B^{-1}\sigma_A)$ evaluated for variable feature sets $A$ and $B$. Graph is rotationally symmetric to first term, $\text{tr}(\sigma_A^{-1}\sigma_B)$. Data is taken from simulation.
Gaussian distributions in the d-dimensional feature space. These observations suggest that, contrary to suggestions in literature, using short fixed-length speech segments for comparing and clustering can be highly inaccurate, even if only one segment is short. Many media, such as news or presentation videos, utilize short segments, and although identifiable by humans, machine clustering often fails.

5.2.2.4 Empirical Solution

The KL2 distance measure is derived under two critical simplifying assumptions [Campbell, 1997]: first, that the MFCC vectors are distributed as a d-dimensional Gaussian, and, second, that the sample means and covariances are perfect estimators of the population means and covariances. The first assumption is necessary to allow a closed form evaluation of a d-dimensional integral. The second assumption eliminates the need to model the effects of the two samples’ lengths on the standard errors of their statistics:
the samples are assumed to be infinitely long with zero standard error. A Gaussian mixture model would be more appropriate, but deriving a closed form KL2 for it presents formidable analytic difficulties. Similarly, in practice, the estimated statistics show increasing error as sample length diminishes. However, the analytic modeling of the impact of standard errors also is challenging, even under the assumption of a single d-dimensional Gaussian. The KL2 measure, then, has likely been used without an understanding of these substantial limitations.

In lieu of an analytic closed-form computation parameterized by the incoming lengths, we investigated a possible empirical solution, which adjusts the KL2 distance, based on a simulated model’s response with identical distributions:

$$KL2'(A,B) = \frac{KL2(A,B)}{KL2sim(|A|,|B|)}$$

Equation 4 KL2’, the adjusted KL2 distance

where $KL2'(A,B)$ is the adjusted KL2 distance for speech segment pair $(A,B)$ with lengths $(|A|, |B|)$, and $KL2sim(|A|,|B|)$ is the simulated KL2 value taken from the results of Experiment 1. Results for this solution are presented in Figure 5.14. With the exception of a few outliers, the KL2’ distance for short feature sets is more comparable with that of long ones for data from one speaker.

5.2.2.5 Results

We have tested our solution for clustering on data sets containing 5 and 29 speakers with significantly different-sized speech (6-189 seconds). Figure 5.18 outlines the data set for one presentation video with six presentations. Results for our empirical
Figure 5.18. Results of Speaker Clustering with and without correcting for length. Shown here are dendrogram excerpts of speaker clustering for one presentation video. Highlighted clusters are the results of comparing between segments of significantly different length. With original KL2 (left), segments of the same speaker but of different lengths do not cluster well. Once adjusted with an empirical correction factor (right), clustering improves. Correct clusters are: $A=\{6$ and $57$ seconds$\}$, $B=\{17$ and $59$ seconds$\}$. Clusters of other matching segments is preserved.

solution are very favorable and include two observations. Firstly, after applying the correction factor, short speaker segments are more reliably clustered with the true cluster to which they belong, thus both validating and correcting the previous shortcomings of the KL2 distance. Two examples are presented in Figure 5.18; without correcting for the differences in lengths, speech segments from speakers A and B do not appear closely clustered or their height indicates dissimilarity (left-hand graphs). After applying the adjusted KL2 measure, these speech segments cluster more reliably (right-hand graphs), and their heights in the dendrograms are more indicative of their similarity. Previously, longer speaker segments clustered more easily together, an effect ascribed mainly to their
length. With the inclusion of the length effect factor, it is possible for shorter segments to correctly cluster earlier with longer ones, if they belong to the same speaker.

Clustering results improved locally on a presentation level (not shown), as well as globally over a series of presentations. We have tested the KL2’ correction with the original KL2 distance, which included covariance and mean terms, as well the modified KL2 distance without the mean proposed in [Li and Dorai, 2004]. Results improved for both, and we note that KL2’ without the mean term performed slightly better in clustering.

5.2.2.6 Speaker Clustering using VQ is not as Effective

There is another alternative to KL useful for clustering. A method of vector quantization can be derived as an approximation to KL2 [Vasconcelos, 2001]. However, clustering based on VQ density estimates tend to exhibit a similar degradation of results for comparisons of speech segments with highly varying lengths from the same speaker. We have designed a correction, and performed a simulation similar to that for KL2, and compared the clustering performance between the original and the corrected VQ
similarity measure. Results (not shown) are again favorable towards a correcting measure. Figure 5.19 highlights the observed distortion in VQ due to variations in sample sizes with a similarity score ranging from $[0.5 .. 0.82]$. It is less pronounced in its absolute value than the distortion observed in KL2 ($[0 .. 45]$), but still high enough to cause clustering of segments with highly varying lengths to fail.

An existing approach presented in [Kinnunen, 2006] aims to optimize VQ-based speaker identification. This algorithm addresses the sample size effect in its computation of the match score, which produces an “average quantization distortion” (AQD). This step divides the cumulative matching score by the number of speaker feature samples, apparently in order to account for sample size.

We have performed a simulation to describe the effect of sample size on clustering performance, and present the similarity matrix in Figure 5.20. While a similar trend to that of KL2 and VQ appears, the distortion along the y-axis is not pronounced ($[0.89 .. 1.02]$), and it does not affect clustering results as heavily. From speaker clustering experiments, we have determined that an additional correction to modified VQ does not improve results. However, since VQ is more costly, we find that our measure of KL2’ is the most effective measure for accurately comparing speaker segments of varying lengths.

5.2.3 Visual Face Index

Next to presentation content, students and their performance are the main focus in student presentation videos. The duration of an individual student’s appearance is relatively short and ranges from five seconds to five minutes. A typical 80-minute
The presentation video contains about 60 student appearances, a large number of which are recurrences. A common task for users of video summaries is to locate the appearance of a particular individual, whether by name or visual cues. Instructors may be interested in searching for a student in order to review their presentation performance, while students are interested in finding their own appearance or that of peers. An effective index into these videos must accommodate the intended use of searching for presenters. In the absence of manually annotated names, a visual index of speaker faces is a viable alternative.

5.2.3.1 Selecting Visual Cues

We rely on automatic speaker segmentation [Chen and Gopalakrishnan, 1998] to create a list of individual speakers. We first extract Mel Frequency Cepstral Coefficients ($MFCC_0 - MFCC_{12}$) from the audio track, and then determine change in speakers by applying the Bayesian Information Criterion (BIC). As a test of the validity of facial indices, we then manually extracted regions from video keyframes which best portrayed the individual presenters, and applied contrast normalization to adjust the highly varying lighting conditions in the recorded video. Automatic extraction of faces was attempted using the Viola-Jones implementation [Viola and Jones, 2002] with partial success, although existing face extractors tended to create too many false positives. In combination with prior work in this area [Wang and Chang, 1997; Cutler and Davis, 2000], an automatic generation of visual speaker indices is feasible.

We have designed four slightly different versions of a visual speaker index (see Figure 5.21). In order of search and retrieval performance, they are: (1) medium close-up
Figure 5.21. Four Versions of Speaker Indices in order of search and retrieval performance. Left to right: (1) medium close-up head and profile shot pairs, (2) medium close-up headshot alone, (3) extreme close-up head and profile shot pairs, (4) extreme close-up headshot alone. Performance is measured by duration and completion rate for a user study search task.

Figure 5.22. User Study Task for Search and Retrieval of a Presenter. Given the visual cue of the person, students must find the appearance in a set of videos.

Task 7: Find the appearance of the person pictured to the left of this text. This student appears at least once in one of the videos. Once found, either place a marker on the timeline, or select the face in the "Speaker List", then hit CONTINUE! If you cannot find this student's appearance, hit SKIP!

head and profile shot pairs (75x75 pixels), (2) extreme close-up headshot alone (50x50 pixels), (3) medium close-up headshot alone (75x75), and (4) extreme close-up head and profile shot pairs (50x50 pixels). As part of our user study, we have measured the effect of each configuration. Students were presented with the task of locating an unfamiliar...
face in a set of videos (see Figure 5.22), each with a visual speaker index as pictured in Figure 5.21. The 129 participants from four sections of our course were randomly assigned one of the four speaker indices. In order to guarantee that the student pictured in the task was not already familiar (i.e. did not appear in the participant’s actual course meeting section), we varied the task material by section.

Effectiveness of each speaker index configuration is measured by the duration and completion rate of the task. Completion rate describes the number of tasks completed versus the total number of tasks. A task is not successfully completed if the participant is unable to find the expected content in the video and skips the question, or if he or she inadvertently completes the task too quickly (suggesting that they did not properly read the question). Results from this task are presented in Section 5.4. The highest performance is exhibited by the speaker index with the most visual detail, namely medium close-up head and profile shot pairs. The success of this configuration cannot clearly be correlated to the use of a medium close-up headshot for the user study task, because the second highest performance for search and retrieval is evidenced by the small headshot, followed by the large headshot. (We speculate that although the frontal/profile pairing gives more information, it may also induce some uncertainty, particularly if an image more closely matches just one of a pair and not the other). Overall, using the best configuration, students required 87 seconds on average to accurately locate a face with a 97% completion rate, followed by 115 seconds for the next best configuration. The complete results are shown in Figure 5.26. This experiment suggests that faces in higher
resolution and pairs of shots in different poses are helpful for faster search and retrieval, but further studies are necessary to evaluate the effect of including several face shots.
5.3 Textual Indices

Parallel to visually summarizing video clips with thumbnails, we use text to summarize audio clips. However, transcripts are not readily available for the presentations, and we cannot make the assumption that every presentation is accompanied by electronic slides.

5.3.1 Textual Feature Extraction

We thus generate transcripts using the IBM ViaVoice ASR. The resulting transcripts are highly imperfect with large Word Error Rates (≈75%) due to several factors. Primarily, the audio quality varies greatly and depends on the individual presenter and the presentation environment. Due to the large number of speakers, it is impractical to apply speech model adaptation, a training process used by speech-to-text systems to capture an individual’s characteristic phonetic patterns and to align them with their textual counterparts. Language model adaptation, a process which builds custom dictionaries and indices of likely term-to-term occurrences, is also unfeasible, as the contents, style, and fluency of the presentations have high variance as well. We therefore use the speech model of this thesis’ author and no custom language model. Experiments with other speech models yielded few improvements (see Chapter 4.2.1.2).

Transcripts with high error rates do not lend themselves to usual text analyses, in which correlations are found between repetition and uniqueness of words and phrases. In Chapter 4.2, we have introduced methods by which highly imperfect transcripts from university lecture courses are filtered by using expected significant index terms extracted
from external course-related sources such as textbooks or web pages. We apply a similar method to transcripts from presentation videos. While we do not have indicators of the specific contents for a given presentation, we do have some knowledge about the overall structure.

Presentations in the domain of our test video database revolve around Engineering Design projects. We have manually generated a list of 30 frequently used words and phrases from the presentation slide titles, and we use them to filter the transcript (see Table 5.1). The resulting “theme phrases” are included in the user interface and provide the equivalent of a table of contents for each presentation (see Figure 3.2 and Figure 3.3). (This task of extracting theme phrases can be automated). The list of theme phrases in Table 5.1 is considered static with respect to the course in which the tool is used. For other domains, outside of the videos we have used, this list of theme phrases could be compiled by cross-referencing frequently used slide headers, or by listing all titles from the presentation slides.

Besides identifying theme phrases, we filter phrases found in the source data of the electronic presentation slides, if available. To this end, each line of text in the slides is used as a phrase. The resulting “topic phrases” are included as an additional index in the user interface and give clues about specific items discussed in the presentation, including names, locations, numbers, etc. Table 5.2 lists examples of topic phrases found in presentation slides. Topic phrases are further refined in the usual way. Methods of stemming and specific stop word removal are applied (see Appendix A for a list of stop words).
Table 5.1. Topic phrases (slide titles) for presentations in the course “Engineering Design”, taken from both ASR and presentation slides.

<table>
<thead>
<tr>
<th>Two word phrases</th>
<th>One word phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>alternative solutions</td>
<td>background</td>
</tr>
<tr>
<td>continuity plan</td>
<td>chart</td>
</tr>
<tr>
<td>design constraints</td>
<td>constraints</td>
</tr>
<tr>
<td>functional requirements</td>
<td>continuity</td>
</tr>
<tr>
<td>future directions</td>
<td>deliverables</td>
</tr>
<tr>
<td>gantt chart</td>
<td>demo</td>
</tr>
<tr>
<td>objective tree</td>
<td>functional</td>
</tr>
<tr>
<td>problem statement</td>
<td>future</td>
</tr>
<tr>
<td>projects goal</td>
<td>goal</td>
</tr>
<tr>
<td>tasks performed</td>
<td>implementation</td>
</tr>
<tr>
<td>team process</td>
<td>limitations</td>
</tr>
<tr>
<td>team development</td>
<td>objective</td>
</tr>
<tr>
<td></td>
<td>prototype</td>
</tr>
<tr>
<td></td>
<td>requirements</td>
</tr>
<tr>
<td></td>
<td>schedule</td>
</tr>
<tr>
<td></td>
<td>solutions</td>
</tr>
<tr>
<td></td>
<td>statement</td>
</tr>
<tr>
<td></td>
<td>tasks</td>
</tr>
</tbody>
</table>

Table 5.2. Example phrases from presentation slides, used as a filter for inaccurate ASR.

<table>
<thead>
<tr>
<th>Example phrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHP (hypertext preprocessor)</td>
</tr>
<tr>
<td>Coming Soon …</td>
</tr>
<tr>
<td>Registered Domain</td>
</tr>
<tr>
<td>Team HIT (the name of the team)</td>
</tr>
<tr>
<td>Dr. Morrison Client</td>
</tr>
<tr>
<td>Ruler prototype (prototype software of a math tool for children)</td>
</tr>
<tr>
<td>Jumping Frog (example of a game)</td>
</tr>
<tr>
<td>Digital I-Pen Mouse (example of a product)</td>
</tr>
</tbody>
</table>
5.3.2 Textual Feature Display

In our user interface, we provide a slider, “Text Context”, which computes the temporal extent within which a phrase is repeated. Words and phrases are initially displayed as horizontal blips, whose width corresponds to the space needed to represent the text, centered on the actual occurrence of the phrase (see Figure 5.23). With increasing text context, similar words are grouped temporally, and blips expand horizontally to mark the duration over which the represented text is used in the video (see Figure 5.24). As frequently used phrases are grouped, less frequent phrases disappear from the text graph, therefore emphasizing the thematic content. The extent to which controls are used depends on the user. We recognize that some users are able to maintain a reasonable overview with a large amount of displayed data, while others prefer to work with smaller amounts.
5.4 Task-Based Experiments and Evaluation

For over 3 years (6 semesters), we have administered user studies, as an integral part of the course work, to measure overall performance changes of video search and retrieval using the VAST MM browser (see Appendix C for a complete list of user studies). We are able to attribute some of the improvements to specific index and user interface elements because changes to VAST MM were made incrementally. In this section, we report on some of the most interesting quantitative observations and evaluations we have made.

Anecdotally, we also observe a significant improvement in the quality of student presentations, when previously recorded videos of prior presentations were available through VAST MM. Students were able to evaluate past performances prior to giving their first in-class midterm presentation, and they were able to evaluate their own and peer performance prior to giving their final presentation. Instructors of the course noted the positive impact of available video resources through VAST MM.

The supervised user study setup in the lab classroom remained identical throughout. Approximately 160 students, the complete enrollment of the first-year introductory engineering design course participated in the user study at the end of each semester. The majority (> 97%) of students are first-year undergraduate engineers, while the remaining students are distributed among senior-level years. Most students fall into an age range of 18-19, with approximately 35% female and 65% male. None of these first-year students repeat the course, and therefore our subjects are unique across semesters. For logistical reasons, students are enrolled in one of four course sections,
which meet during different schedules throughout the week: Monday afternoon from 1:00pm to 4:00pm, Monday night from 4:20pm to 7:20pm, Tuesday afternoon from 1:00pm to 4:00pm, and Tuesday night from 4:20pm to 7:20pm. During the course of the semester, students hold midterm presentations, which are video-recorded and made available permanently at the time of the user study. While students are well aware of the project presentations within their own course section, they are generally unfamiliar with those from the other three sections.

Supervised in-class user studies are conducted in small groups of 5-14 subjects, requiring between 30 and 60 minutes of time depending on individual students’ speed for completing tasks. All user studies are completed on lab computers, which, with the exception of Fall ’07 remained the same throughout the preceding five year period: Dell Precision 530 workstations with Intel Xeon 1.7GHz processors, 1GB RAM, and 17” LCD panels. No upgrades of any kind took place until in Fall ’07, the equipment was replaced by Apple Mac Pro 8-core workstations with 2GB RAM, and 23” LCD panels.

Each group of participants was collectively introduced to the VAST MM browser in a 15 minute hands-on session with the system. Students were also given instructions on the user study setup and the post-study survey. During the study, students were free to ask the administrator any questions and give immediate feedback regarding unexpected behavior as necessary. Detailed logs of interaction with the tool were collected automatically for each user (students used unique logins to access the tool).
Table 5.3. User study tasks for VAST MM browser evaluation.

<table>
<thead>
<tr>
<th>Task</th>
<th>Example formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locate familiar content</td>
<td>“Find your appearance during your team’s presentation”</td>
</tr>
<tr>
<td>Locate familiar content</td>
<td>“Find the beginning of your team’s presentation”</td>
</tr>
<tr>
<td>Locate familiar content</td>
<td>“Find your team’s discussion on functional requirements”</td>
</tr>
<tr>
<td>Locate unfamiliar content</td>
<td>“Find the presentation related to building a greenhouse for a local school”</td>
</tr>
<tr>
<td>Summarize unfamiliar content</td>
<td>“Locate the presentation in video X between time Y-Z. Summarize the presentation using only the displayed keywords and phrases”</td>
</tr>
</tbody>
</table>

With only few exceptions, user study tasks repeated steadily across semesters, in particular search of familiar and unfamiliar content, and summarization tasks (see Table 5.3). For cross-semester evaluation, we also ensure that comparisons are made only for those subsets of data, which were collected under similar conditions. For example, our benchmark evaluation of a difficult search task (search for unfamiliar content), we only include data collected from studies performed in the classroom and with the availability of actual video. Data collected from participants completing the study in their dormitory was excluded.

5.4.1 Structure in Video

Structure in information helps in the dissemination of ideas and in the understanding of otherwise massive and complex concepts. Especially when information is dense, as in a textbook or a technical paper, structure in the form of chapters, sections,
subsections, etc. are imposed to better organize the material. Tables of contents and various alphabetized indices serve as fast retrieval methods.

Structuring information in videos can benefit dissemination of its contents similar to how it works with books. This is particularly true of unstructured, i.e. unedited videos, such as presentation videos. In their unedited state, presentation videos feature several presentations back to back. A helpful structural cue for such a video would indicate transitions between presentations. With the addition of this cue, we are able to better describe a video’s content in a form equivalent to book chapters (presentations) and sections (individual presentation slides).

To evaluate the added benefit of providing structure cues in presentation videos, we have built an audio-based detector to identify the clapping of hands, which generally marks the end of a presentation. The browser user interface was then modified to provide a visual mark at each presentation transition so detected. Finally, we administered a user study to measure the utility of this information.

In this study, we used an ablation method. Of the four most relevant sources of information (actual video, keywords and key phrases, keyframes for each “scene” of near-constant visual similarity, and audio-assisted presentation segmentation), participants were presented with only two of these cues to complete the user study; the remaining two cues were disabled. If we consider presentation segmentation and keyframes to be structural cues, and actual video and text to be unstructured, we identify 4 interesting pairs to test: (1) structured: segmentation + keyframes, (2) semi-structured:
Table 5.4. Evaluation of search tasks performed with four variations of a cue-rich browser: Keyframes + Presentation Segmentation, Keyframes + Text, Video + Pres. Seg., and Video + Text. Users who had access to features based on structure (Keyframes and Pres. Seg) generally finish search tasks in the least amount of time. Table cells colored in green emphasize the best, while those colored in red point out the worst results.

Participant sample size:

<table>
<thead>
<tr>
<th></th>
<th>Keyframe</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pres. Seg.</td>
<td>40</td>
<td>39</td>
</tr>
<tr>
<td>Text</td>
<td>37</td>
<td>40</td>
</tr>
</tbody>
</table>

Task: “Find the beginning of your first (or only) appearance in which you spoke during the presentation”

<table>
<thead>
<tr>
<th>Duration (sec)</th>
<th>Keyframe</th>
<th>Video</th>
<th>Completion</th>
<th>Keyframe</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pres. Seg.</td>
<td>31.78</td>
<td>72.43</td>
<td>Pres. Seg.</td>
<td>90%</td>
<td>95%</td>
</tr>
<tr>
<td>Text</td>
<td>38.94</td>
<td>87.80</td>
<td>Text</td>
<td>94%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Task: “Find the beginning on your team's discussion on Functional Requirements”

<table>
<thead>
<tr>
<th>Duration (sec)</th>
<th>Keyframe</th>
<th>Video</th>
<th>Completion</th>
<th>Keyframe</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pres. Seg.</td>
<td>57.38</td>
<td>125.77</td>
<td>Pres. Seg.</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Text</td>
<td>68.14</td>
<td>106.58</td>
<td>Text</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>

Task: “Find the presentation on the <TITLE> in any of the provided videos”

<table>
<thead>
<tr>
<th>Duration (sec)</th>
<th>Keyframe</th>
<th>Video</th>
<th>Completion</th>
<th>Keyframe</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pres. Seg.</td>
<td>190.24</td>
<td>348.70</td>
<td>Pres. Seg.</td>
<td>68%</td>
<td>64%</td>
</tr>
<tr>
<td>Text</td>
<td>217.69</td>
<td>353.22</td>
<td>Text</td>
<td>46%</td>
<td>46%</td>
</tr>
</tbody>
</table>

Task: “Summarize presentation X”

<table>
<thead>
<tr>
<th>Duration (sec)</th>
<th>Keyframe</th>
<th>Video</th>
<th>Completion</th>
<th>Keyframe</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pres. Seg.</td>
<td>59.95</td>
<td>126.44</td>
<td>Pres. Seg.</td>
<td>94%</td>
<td>99%</td>
</tr>
<tr>
<td>Text</td>
<td>48.59</td>
<td>71.30</td>
<td>Text</td>
<td>97%</td>
<td>90%</td>
</tr>
</tbody>
</table>

segmentation + video, (3) semi-structured: text + keyframes, and (4) unstructured: text + actual video.
Participants were randomly assigned one pair of cues, and we ensured that the distribution was close to even. All participants in the user study were assigned the same search and summarization tasks. Results from our study suggest a strong correlation between time required to complete search tasks and the availability of structural cues (see Table 5.4). In most cases, unstructured cues demand 100% more time for task completion. In the particularly difficult task of finding previously unfamiliar content, structural cues also help significantly with successful task completion. In each task, minimum time required for completion and maximum completion score involved keyframes, while maximum time required and minimum score always involved actual video.

We also note that summarization is the only task that clearly benefits from the availability of text cues. However, this is also the only task that does not involve search. Interestingly, the presence of actual video still increased the time required to complete this type of task.

5.4.2 Availability of Actual Video is Counterproductive

Video is rich in redundant information, an observation that is exploited in the design of MPEG compression. In particular in the visual domain, many video scenes present the same information with only small changes in activity. In a series of 3 semester user studies, we have evaluated how this redundant information impacts the time required to complete directed search and retrieval tasks. For the completion of their user study, all participants had access to the same version of the VAST MM browser featuring
Effect of Availability of Streaming Video

Figure 5.25. Availability of Video is Counterproductive: Average task duration for user studies with and without the availability of video. Without video, participants complete tasks significantly faster (up to 74% faster).

Table 5.5. Task duration and completion as a function of availability of video playback. When video is not available, participants complete tasks significantly faster at equivalent completion rates. Table cells colored in green emphasize the best, while those colored in red point out the worst results.

<table>
<thead>
<tr>
<th></th>
<th>Actual video available</th>
<th>Actual video not available</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Completion</td>
<td>Duration (sec)</td>
</tr>
<tr>
<td>Spring 2005</td>
<td>90%</td>
<td>84.81</td>
</tr>
<tr>
<td>Fall 2005</td>
<td>89%</td>
<td>126.78</td>
</tr>
<tr>
<td>Spring 2006</td>
<td>89%</td>
<td>131.71</td>
</tr>
</tbody>
</table>

browseable video summaries (keyframes). However, we enabled actual video only for half of the participants. Tasks remain the same throughout.

We observe that browsable video index cues are sufficient for successfully completing the search tasks, as demonstrated by a comparable completion measure (see Table 5.5). Time required to complete the task, however, is significantly lower (by 20% -
Figure 5.26. Face Indices Save Time: Average time required for completion of face identification task. On average, the more visual information is available, e.g. in the form of two large face images, the faster a task is completed.

40%) when actual video is not available (see Figure 5.25). We can attribute this difference to the “familiarity effect”. Anecdotally, we observe that when actual video is available, students tend to make use of this more familiar medium. When actual video is disabled, students are bound to explore the browsable video summaries.

5.4.3 Face Indices Save Time

When videos feature several actors, it is not only important to represent is being communicated, but also who is communicating and how often they appear. A presentation video may contain 20 or more students, speaking in variously sized time intervals and repetitions. To gain a better view of the actors in the video, we have introduced a visual face index (see Chapter 5.2.3), which can be used to locate presentation segments based on speakers. The face index was implemented in four
Table 5.6. Average task duration and completion in user studies performed in dormitories and in-class. Students are much more focused on completing tasks quickly in class.

<table>
<thead>
<tr>
<th></th>
<th>Prep. Time (sec)</th>
<th>Prep. Actions</th>
<th>Completion</th>
<th>Duration (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Class</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2006</td>
<td>949</td>
<td>902</td>
<td>96%</td>
<td>107.18</td>
</tr>
<tr>
<td>Dorm</td>
<td>323</td>
<td>108</td>
<td>89%</td>
<td>208.13</td>
</tr>
<tr>
<td>In Class</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2007</td>
<td>668</td>
<td>492</td>
<td>97%</td>
<td>148.38</td>
</tr>
<tr>
<td>Dorm</td>
<td>76</td>
<td>33</td>
<td>76%</td>
<td>334.90</td>
</tr>
</tbody>
</table>

variations in which a speaker was represented by (1) an extreme close-up headshot alone (50x50 pixel), (2) an extreme close-up head and profile shot pair (50x50 pixels), (3) a medium close-up headshot alone (75x75 pixel), and (4) a medium close-up head and profile shot pair. In our experiment, we have measured the time required to complete two tasks: finding themselves in the video collection, and finding a different student who they are unfamiliar with. On average, required search time decreases with increasing information content, in the order in which the face combinations are listed above. However, on an individual task basis, slight variations are apparent (see Figure 5.26). The experiment demonstrates that face indices are most effective when they provide more visual information. We speculate, however, that there exists a limit to the benefit of additional information when individual face images become too large and the number of faces displayed in the UI too small.

5.4.4 In-class Use is More Efficient

Over a period of two semesters, we have evaluated the effect of participants completing their user study in a formal classroom setting versus a familiar dormitory
setting. While we expect students to conduct themselves very differently in these two settings, the measurable effect is not clear. A comparison shows that students in a classroom setting are likely to spend more preparation time familiarizing themselves with the browser. We note, however, that in-class user studies are also conducted with more rigorous organization, and some preparation is mandatory at the time when the administrator introduces the study. We observe a direct correlation between preparation time, and successful task completion and speed (see Table 5.6), discussed further in Chapter 5.4.6.

The dichotomy between in-class and in-dorm completion time is largely due to a shift in usage pattern closely related to the discussion in Chapter 5.4.2. Students who completed the study in dormitories commonly revert to actual video and do not take advantage of the summaries. In-class participants, however, were given an interactive demonstration of the tool, and were therefore prepared for use of the browsable summaries. Several other indicators confirm this hypothesis, including increased usage of keyframe in-class versus in-dorm, significantly more interaction with UI customization parameters in-class, etc.
5.4.5 VAST MM Improves Over Time

Finding previously unfamiliar information is one of the most difficult video search tasks. For example, a verbal query for the desired content can prove unsuccessful if there is a mismatch in vocabularies. Likewise, dissimilarities between a user’s formulation of a visual query and the actual visual contents of a video can be equally misleading. For example, a “musical device for people with disabilities” can be interpreted as the picture of an iPod, drum set, jukebox, home stereo, etc., when in fact it can be a computer program that specially interprets a standard PC keyboard.

In our extensive user studies, we placed great emphasis on the performance improvement of such tasks over time. With constant updates to index cues and various search and user interface features in VAST MM, we were able to improve average completion times from 57% to 97%, and to lower average times required for these tasks from 436 seconds to 128 seconds (see Figure 5.27). We note that for these results, we only consider user studies performed under similar conditions, i.e. in-class and with the availability of actual video.

5.4.6 Users Improve Over Time

We anticipated that a user’s familiarity with the VAST MM browser would lead to improvements in task performance. Not unlike the experienced web searcher, who selectively searches for text and images using various query styles, a user of VAST MM should exploit different modalities to speedily complete a search task. While we have not tested this hypothesis extensively, we find a correlation between familiarity and
5.4.7 Cues improve Time of Task Completion

We determine what features of the VAST MM browser are most effective for successful completion of user studies. Under the assumption that the data obtained for the various features (variable $X$) in conjunction with task duration (variable $Y$) follows a parametric distribution, we apply the Pearson product-moment correlation coefficient. PMCC is a common measure of correlation between two variables $X$ and $Y$. The statistic
is defined as the sum of the products of the standard scores of the two measures divided by the degrees of freedom \((n-1)\):

\[
r = \frac{\sum z_x z_y}{n-1}
\]

Pearson's correlation depends on the degrees of freedom \((df)\) in the data, which is equal in our case to two less than the number of subjects: \(df = N - 2\). Using a critical value table, we can find an entry at the intersection of degrees of freedom, with the desired alpha value that determine significance level. The entry noted in the table is the minimum correlation coefficient, \(r\), needed to prove significance at that alpha level.

In evaluating useful cues, we find significant correlations to task duration for the following features at \(\alpha = 0.05\). In all user studies:

- Skimming of high-quality keyframes
- Viewing actual video
- Zoom activity (changing amount of content displayed)
- Low zoom, i.e. zoomed out (displays more information)

In some of the user studies we found:

- Scene segmentation activity, i.e., number of thumbnail keyframes displayed (displays more visual changes)

### 5.5 Conclusions

Through differential user studies we can formalize a set of features towards an improved content browser for unstructured video. Generally, cues to organize content into a book-like structure, such as segmentation into content units on various levels, help
significantly in search and retrieval. Similarly, multi-modal summarization and representation cues, such as representative keyframes, face indices of actors in a video, and key phrases emphasizing themes, are very helpful in indexing browsable content. While actual video is counterproductive in search-related tasks, it is clearly indispensable for viewing content, and therefore, we retain it in the VAST MM browser.
In this chapter, we introduce several indexing approaches common across all genres of video. This includes temporal alignment of transcript text to speech when such alignment is not provided by speech-to-text engines, keyword user interfaces, video annotation tools for interactive personal and public commentary and feedback, and a taxonomy browser for arbitrarily defined categories. While there exist many cross-genre tools for video indexing and organization, we found these most relevant for VAST MM.

6.1 Audio-Text Alignment

Speech transcriptions for semi-professional video productions, such as lectures, have traditionally been a manual task performed by commercial transcription services. Depending on the invested effort, such transcripts are either perfect or approximate. Not in all cases are such transcripts annotated with time stamps. With the introduction of ASR systems, similar approximate transcriptions can be produced, especially when speaker
and language models are appropriately created. In a classroom environment with more than 100 students exhibiting a wide variety of accents and speech qualities, training becomes an infeasible task. Relying on commercial software packages such as IBM ViaVoice and Dragon NaturallySpeaking is an easy and inexpensive alternative, but again, temporal annotation is not provided. Untrained use of ASR systems in a sub-optimal recording environment comes with a severe trade-off in recognition quality and lack of text-to-speech alignment. Word alignment error rates (WER) are commonly as high as 75%. WER in this context is defined as:

\[
WER = \frac{|insertions| + |deletions| + |replacements|}{|referenceText|} \times 100
\]

Despite the low accuracy, we have shown in Chapter 4.2 that for keyword indexing and searching, this quality is sufficient.

The missing temporal alignment data from automatic transcripts limits the search and retrieval capability of multimedia browsers. In general, a cheap ASR transcript combined with a cheap post-processing method is likely a cost effective approach. We address the problem of aligning highly imperfect transcripts to their original speech signal by relying on classes of phonemes, which are recognized with high accuracy. We apply our approach to lecture and student presentation videos for indexing and archiving.

### 6.1.1 Approach

Our approach addresses several shortcomings of the speech and text in our data set. Speech data is taken from audio sources with varying qualities in compression and in
Table 6.1. Top: manual (perfect), Bottom: automatically generated (highly imperfect) transcriptions. Timestamps precede the phrases. Highlighted are correctly identified words, correctly identified stems, and incorrect, but similarly sounding words.

<table>
<thead>
<tr>
<th>1950: for a way in which people can tell a video is different from another video based on the fact that they are simply different faces you have to be aware of these</th>
<th>1960: of things you wanna retrieve according to some sort of deep structure formalism so for example range of color or</th>
</tr>
</thead>
<tbody>
<tr>
<td>1950: for a way content the fact that can be captured a</td>
<td>1960: with and one of the true accounting deferrals are indeed structure plan live on or example range of colors but</td>
</tr>
</tbody>
</table>

recording environment. The audio taken from lectures is obtained from highly compressed videos, in which the instructors' audio quality is roughly constant throughout. However, the audio taken from student presentations is taken from less compressed videos (DV tapes), but where the recording qualities are not constant. Besides these highly varying speech and language qualities, the many different speakers (students) have many different presentation characteristics.

Transcripts were obtained through the IBM ViaVoice ASR engine without any specially designed language or speaker models. The resulting text exhibits a typical WER of over 70% (computed from edit distance) with the number of correctly identified words below 40%. Depending on an individual speaker's characteristic, a transcription can be reasonably accurate or filled entirely with noise. In some cases where the recording quality or the speed of speakers' presentation changes dramatically from the norm, we observe cases where the ASR engine skips several words at a time. An example of a typical transcription is presented in Table 6.1.
Figure 6.1. Overview of Speech-Text Alignment for a sample phrase (without errors). Audio is filtered and selected phonemes are extracted using formant estimation or dominant frequency readings from a spectrogram. Temporally unaligned text (perfect or imperfect) is converted to phonemes. Alignment is performed using dynamic programming edit-distance transformation.

Due to these varying characteristics, an approximate temporal alignment between text and speech using easily obtained parameters, e.g. signal strength, is not favorable.

We have observed through experimentation, that while transcriptions contain significant word errors, vowels and fricatives tend to be recognized with high accuracy and can still be found in the incorrectly recognized words. Our approach takes advantage of this observation, and performs text to speech alignment based on the selective subset of monophthongs (single vowels) and fricatives (see Figure 6.1).
Table 6.2. (top) Phonemes detected in speech and text; (bottom) Rules of substitution for phonemes not specifically detected in speech, but exhibiting similarities.

<table>
<thead>
<tr>
<th>Monophthongs:</th>
<th>IY (bee_t), IH (bi_t), EH (be_t), AE (ba_t), AH (above), UW (boot), UH (book), AA (father), ER (bird), AO (bought)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fricatives:</td>
<td>SH (assure), S (sign)</td>
</tr>
<tr>
<td>Diphthongs:</td>
<td>AW (ou) → AH, AY (fīve) → AH, EY (day) → AE, OW (crow) → UH, OY (bov) → AO</td>
</tr>
<tr>
<td>Fricatives:</td>
<td>Z (resign) → S</td>
</tr>
<tr>
<td>Affricates:</td>
<td>CH (church) → SH</td>
</tr>
</tbody>
</table>

We summarize our method here. We first filter audio data for regions that do not correspond to speech, or are unlikely to be transcribed by the ASR engine due to poor audio quality. (Although we easily detect silences, the ASR transcript does not indicate them.) Phoneme detection is performed on this filtered signal, resulting in a temporally accurate list of speech phonemes. We tokenize the unaligned transcript into phonemes using a dictionary. Speech phonemes are then aligned to text phonemes using dynamic programming, resulting in a temporal alignment which can be mapped to the original transcript. Details on our method now follow.

6.1.1.1 Phoneme Extraction from Audio

The goal of phoneme extraction is to assemble a list of likely occurring phonemes in the audio signal (see Table 6.2), which can later be aligned to text phonemes.

Monophthong detection takes advantage of the characteristic stable voicing for vowels, which can be estimated by formant frequencies. The Matlab-based toolkit introduced in [Digital Bubblebath, 1996] models the vocal tract using an autoregressive
model of the speech signal. Peaks in the frequency response correspond to the resonant frequencies of the vocal tract, or formants.

Using a table of expected frequency values of formants $F_P = (f_1, f_2, f_3)$ for each classified phoneme $P$, the closest match to an unclassified phoneme is determined by the difference ($F_D = F_C - F_P$) between computed formant vector ($F_C$) and classified formant vector ($F_P$), weighted to emphasize formant $f_1$, $F_W = F_D \times (2,1,1)$, because it best discriminates between phonemes. The algorithm selects the phoneme with the minimum Euclidean distance of all candidate vectors $d_{\text{min}} = \min(\text{norm}(F_{W1}), \text{norm}(F_{W2}), ...)$, the value of which is then compared to an empirically determined threshold to accept or reject the phoneme (a distance value closest to zero determines the best match).

Detection of fricatives is highly dependent on the distribution of energy in frequency bands illustrated by spectrograms. We use the expected distributions of energy among frequency bands to detect the fricatives SH, and S. For a given window of speech signal, we select the maximum value of normalized cumulative energy in the expected frequency bands: SH/CH = [2500-3000 Hz], S/Z = [3000-4000 Hz], all others = [300-2500 Hz].

Phoneme detection is performed on small windows of the audio signal of 1/30th of a second in length. This window is intentionally smaller than the average duration of a phoneme in particular vowels. In a final step, neighboring phonemes of the same type are merged, without smoothing.
6.1.1.2 Phoneme Extraction from Text

In this step we generate a collection of phonemes from the unaligned transcript. While pronunciation of words depends largely on dialect, it is infeasible to tune the phonetic dictionary as a dialect model, in particular when an audio track features 30 or more speakers in short intervals. We assume pronunciation for American English, and make use of the CMU Pronouncing Dictionary with over 125,000 words and their phonetic transcriptions [CMU Pronouncing Dictionary].

Text is segmented into words, which are then represented by their phonemes. Words not found in the dictionary are shortened to find the closest stem. Words that pronounce numbers are converted to their verbal counterparts, while treating the pronunciation of isolated digits differently from that of compound numbers, e.g. “1-0-0 = one zero zero” versus “100 = one hundred”. We apply a set of rules for several phonemes which are not detected during speech phoneme extraction, but which share phonetic features with the identified monophthongs (see Table 6.2). The final representation of a word includes only phonemes that can be detected by speech.

6.1.1.3 Alignment

We perform alignment globally between the sets of phonemes from speech and from text using the edit-distance dynamic programming algorithm, similar to the alignment task between two DNA sequences. Our dynamic programming edit-distance implementation aligns text phonemes to a larger set of speech phonemes by incurring costs for copies, deletions, insertions, and replacements, but not transpositions. Because
Table 6.3. Speech-text alignment audio sources, statistics, and results. Lecture video features one speaker (male) with constant audio quality. Student presentations feature many speakers and audience with highly varying audio and speech qualities.

<table>
<thead>
<tr>
<th>Video; Transcription generation</th>
<th>Features</th>
<th>Length</th>
<th>Avg. Matching Error (sec)</th>
<th>Speech phonemes</th>
<th>Text phonemes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Lecture; manual</td>
<td>Single speaker, one long break (504 sec)</td>
<td>1:48:21</td>
<td>3.9</td>
<td>64265</td>
<td>27645</td>
</tr>
<tr>
<td>B Lecture; automatic</td>
<td>Single speaker, one long break (504 sec)</td>
<td>1:48:21</td>
<td>7.7</td>
<td>64265</td>
<td>21608</td>
</tr>
<tr>
<td>C Student Presentation, automatic</td>
<td>31 speakers, 6 Q&amp;A sessions of varying durations (30 – 300 sec)</td>
<td>1:15:12</td>
<td>6.43</td>
<td>54537</td>
<td>16248</td>
</tr>
<tr>
<td>D Student Presentation; automatic</td>
<td>10 speakers, 2 Q&amp;A session of durations 60 sec and 185 sec</td>
<td>0:22:32</td>
<td>26.73</td>
<td>12596</td>
<td>4520</td>
</tr>
</tbody>
</table>

of the significantly redundant set of speech phonemes, the cost of incurring copies and deletions (-1) are the same, while replacements and insertions are assigned a cost of +1.

A typical set of speech phonemes for 60 minutes of audio contains up to 45,000 elements, while the equivalent set of text phonemes contains up to 15,000 elements (see Table 6.3). Once completed, time codes from individual phoneme matches are assigned to their original words, thus producing the temporal text to speech alignment for the user interface.
6.1.2 Experimental Results

We have conducted experiments with 3 speech files and with 4 transcriptions. Three of the 4 transcriptions were highly imperfect and automatically generated, but one was perfect and manually generated. Speech files were taken from lectures and student presentations (see Table 6.3). We note that alignment is accurate within a reasonable error margin, and in subsequent user studies we observed that it was sufficient to search a video stream. On average, more than 60%, 75%, 90% of all words are aligned correctly within a 10, 20, 30 second error margin, respectively. Figures 6.2(a-e) illustrate the phoneme alignment error for the data sets. The significant jump in Figures 6.2(a) and Figures 6.2(b) are due to a silence break of more than 8 minutes in the extended lecture of 108 minutes. Similar spikes can be found in Figures 6.2(c), where five Q&A periods between presentations cause phonemes close to the silenced break to be misaligned. The noticeably large error in Figures 6.2(d) is due to a number of factors related to speech quality, including increased speed of the student’s speech and volume, both leading to a low (20%) rate of transcribed words. Also in Figures 6.2(d), in the following presentation after a 58 second Q&A break, speakers exhibited strong accents, resulting in a larger than usual WER. The combination of these factors causes the large error in phoneme alignment of up to 80 seconds.

Our experimental setup and calculation of phoneme alignment error causes an additional source of error that was not precisely measured here. Ground truth time alignment is manually inserted into the transcripts. For the manual transcription, time
codes are placed every 10 seconds, and for automatic transcriptions at varying points between 10 and 30 seconds. However, timestamps of words between these inserted time markers have to be interpolated. Clearly, speech does not exhibit constant temporal intervals between words – but this linear approximation is required because alignment
cannot be clearly determined, due to missing and falsely identified phrase segments. Overall the results are promising considering that large portions of text are unrecognizable, and only their phonetic constructs hint at their intended meaning. Anecdotally, it also outperformed other available tools such as [Martin, 2004].
6.2 Keyword and Key Phrase UI

Search and retrieval of video content rely heavily on cues drawn from text, either in the form of speech through automatic speech recognition (ASR), or from writing through optical character recognition (OCR). Existing multimedia content browsers for news [Campbell et. al., 2006] and sports use textual cues mostly from ASR, which is reasonably accurate for edited videos with high-quality audio. However, ASR transcripts from unedited videos, such as lecture and presentation videos recorded in noisy environments, exhibit higher word error rates (WER) of up to 75% (see Table 6.4 and Chapter 4.2.1). Selection and relevancy ranking of key terms from these transcripts is difficult. Information retrieval methods, such as TF-IDF fail on the raw data, because most of the unique yet incorrectly identified terms are salient.

6.2.1 Text Selection and Ranking

We apply relevant external indices to filter raw transcripts. We have shown in Chapter 4.2 that course textbook indices are effective filters for lecture videos. Raw transcripts from student presentation videos can best be filtered with text content from presentation slides or other course material. We first extract all text from presentation slides, including bullet points, paragraphs of prose, titles, captions, etc. Each of the resulting text entities is then filtered for meaningful words and phrases using the WordNet [Fellbaum et. al., 1998] lexical database. Stemming takes place only to remove plural senses, because fully stemmed terms cannot be queried in WordNet. (From here on we use “phrases” to also denote “words”). We build three sets of relevant phrases:
Table 6.4. Raw automatically generated transcript. Valid terms are highlighted in bold. Index terms are underlined. This presentation describes the design of a wheelchair swing.

... preliminary research and design the river begun similar solutions the with and then vote in the doing next few weeks for work and the impact on a long haul that we're given was that children who one any the use on swings of modern this lens without being transferred to standards went subject of why this suit bill the slaying that eight of child who's in los share of would be able to swing and without been transferred to slight and if they're strong enough case also be able to of self propelled ...

Set A: All single terms that are not stop words are extracted into a separate set, including those which can form phrases.

Set B: All WordNet phrases in the text entities are identified by iterating over each word and resolving the longest possible phrase starting with this word. In this iteration, we do not remove stop words, which are helpful in identifying proper phrases, e.g. “Statue of Liberty”.

Set C: Phrases that do not exist in the WordNet database are extracted by identifying all WordNet-resolvable terms and selecting all consecutive words in-between these known words. Oftentimes these include named entities and technical terms, e.g. “... met with Dr. Topsy Krets to discuss ...”.

The combination of these three sets is applied as a filter to the raw imperfect transcripts. The resulting list of matching and time-aligned terms serves as a searchable index into video content. We note that it is very difficult to identify complete sentences, since inaccurate automatic transcriptions not only lack punctuation, but also rarely contain grammatically valid phrases. Alternatively, external indices can be used as text
indices without cross-referencing them to text-from-speech; however, the cross-referencing step is necessary to resolve phrases in specific locations in long video sequences.

We assign to each identified phrase a numerical rank $R$, which is used to emphasize text differently in the browser interface (see Figure 6.3). Terms considered unique or very descriptive are ranked higher, whereas more common terms are ranked lower. We use the following heuristic rules:

Rule 1a, Named entities: If the phrase does not exist in the WordNet database (terms from Set B), then the phrase’s rank $R$ is equal to its number of words times a constant, $T$. $T$ is a numerical value which can be adjusted to emphasize the weight of a named entity in comparison to the measures used to rank terms from other sets.

Rule 1b, Uniqueness: Resolve the number of synonymous senses

\[ numSenses = synonyms(P) \]

for WordNet phrase $P$. Less specific phrases have higher values.

Rule 2, Specificity: Using WordNet’s hypernym sense querying, establish the distance of a phrase $P$’s sense to the root sense (e.g. a noun’s root sense is “entity”):

\[ distRoot = path(P, root) \]
The more distant a phrase is to the root, the more specific and descriptive it is. Without performing sense disambiguation due to highly noisy text data, we use the most common word sense defined in WordNet for resolution.

**Rule 3, Nominality:** If the phrase contains only nouns, boost the rank of the phrase, because pure noun phrases are generally highly specific:

\[
\text{nounEmphasis} = 3 \cdot |\text{nouns}|
\]

**Rule 4, Phrase weight:** The final rank of a phrase is a weighted combination, which relates \( \text{numSenses} \) inversely to \( \text{numWords} \) and \( \text{distRoot} \):

\[
R = \frac{\text{numWords} \cdot \text{distRoot}}{\text{numSenses}} + \text{nounEmphasis}
\]

**Rule 5, Temporal locality:** The user interface separately allows the user to cluster equivalent phrases within a variable temporal vicinity via a slider, resulting in extended phrase ovals (see Figure 6.3). When so clustered, the weight of a phrase increases proportionately to its temporal extent.
6.2.2 Interface and Evaluation

We have compared two user interfaces (see Figure 6.3), one with and one without ranking of text. In both interfaces, phrases are automatically time aligned and placed in a panel with a greedy algorithm, filling space towards the top first. In the interface featuring text ranking, color and vertical position are used to emphasize text of varying weight. Similar phrases are grouped together into a shared blip within a user-controlled time context interval. This interactive setting expands blips horizontally to mark the duration over which the represented text is used in the video.

We have evaluated our method of phrase selection and ranking in a user study with 313 students over two semesters. Students were presented several tasks related to search and retrieval, including summarization of unfamiliar presentations. In this task, students are presented only with the filtered phrases for a presentation from which they must infer the video’s content. They are then asked to select a fitting description of the video from a multiple-choice menu. We measure performance by the duration of a task and the accuracy of the response. Each interface was used in one of two consecutive semesters. Results show that in the presence of text ranking (see Table 6.5, Task 2), accuracy of responses increases and required time per task decreases over the alternative interface without text ranking (see Table 6.5, Task 1).

We have also performed a separate user study with 129 students and 240 summarization tasks, in which students were required to write their response in freeform English. We note that the duration of this task increased to 88 seconds due to the lack of preset responses, but accuracy remained at a high level (see Table 6.5, Task 3).
Table 6.5. Evaluation of summarization tasks with and without text ranking and pre-defined multiple choice answers, and with text ranking and articulation of response.

<table>
<thead>
<tr>
<th>Answer score (measured from pre-defined answers)</th>
<th>Task 1: Without Rank</th>
<th>Task 2: With Rank</th>
<th>Answer score (measured by manual inspection of answers)</th>
<th>Task 3: With Rank, freeform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct</td>
<td>67%</td>
<td>87%</td>
<td>Correct (well articulated response)</td>
<td>82%</td>
</tr>
<tr>
<td>Semi-correct (pre-defined answer was slightly out of context)</td>
<td>20%</td>
<td>10%</td>
<td>Semi-correct (used some key concepts, but slightly out of context)</td>
<td>15%</td>
</tr>
<tr>
<td>Incorrect</td>
<td>13%</td>
<td>3%</td>
<td>Incorrect</td>
<td>3%</td>
</tr>
<tr>
<td>Duration</td>
<td>51.47 sec</td>
<td>41.85 sec</td>
<td>Duration</td>
<td>87.98 sec</td>
</tr>
</tbody>
</table>
6.3 Annotation Tool for Unstructured Videos

Information retrieval on the Internet has long become more dynamic with the introduction of forums and web-logs that allow user annotations. Multimedia retrieval should also take advantage of the same type of user interaction. However, when attaching a comment to a video, the annotation should be temporally localized (see Figure 6.4). This is particularly true of long video sequences with a plurality of content – comments are more meaningful if they address specific temporal localities.

For lecture and presentation videos, feedback linked to audio-visual content can serve the needs of two audiences: instructors and students. For lecture videos, instructors reviewing a day’s lecture can amend lecture material and annotate points not otherwise emphasized during a class. For example, a post-lecture clarification in the form of a text comment placed at the appropriate location in the video can be more powerful than an e-mail sent to students, because it preserves the context of the lecture, and does not require cross-referencing of e-mail and lecture material. In a case of pre-recorded lectures used over a series of semesters, instructors should have the ability to annotate videos and effectively communicate any modified organization, for example, when a recorded topic should be skipped. From a student’s perspective, these annotations can serve as a context-driven discussion forum where clarifications and questions are best linked to the original lecture content.

Likewise for student presentation videos, the feedback dialog would benefit similarly from time-specific annotations. Instructors who evaluate and grade presentation
Figure 6.4. Bookmarks and Annotations in the User Interface appear in a desired temporal location. The orange track towards the top shows a personal bookmark visible and editable only by the current user logged into the VAST MM browser. The orange track towards the bottom shows several locked public annotations notes from another user, as well as a new note in the process of being created. Public annotations can be viewed by everyone, but can only be edited by their owner. Annotations available for editing are emphasized with brighter background and border colors than locked notes. In this figure, the note in the bookmark track and the center note in the annotation track are editable, while six other notes in the annotation track are locked because they have been created by a different user. The bright yellow text entry box in the annotation track is used to create/modify/delete notes.

Performance can use video annotation tools to better highlight effective and ineffective approaches. Instead of lengthy and oftentimes confusing descriptions and directions, annotation notes attached to the appropriate location in the video are simpler and more effective means of making audio-visual examples (“good” vs. “bad” approaches), whether they are meant for feedback, or grading as an archived learning library. While annotations are usually implemented as text-only notes, it is not implausible that audio
notes are more effective for certain video material, for example language courses with speech training.

Annotations supersede most current means of communicating comments in audio-visual content. Personal means of organizing and tracking content in the form of bookmarks already follow a similar approach. But, instead of bookmarking an entire video, personal bookmarks in VAST MM exist in the form of temporally aligned notes. We have made available such personal bookmarks and public annotation features for lecture and presentation videos. Survey results show that these additional features for organization and interaction resonate well with students, although we have done no formal studies.
6.4 Taxonomy Browser via 3D cylinder UI

With the increasing size of a video library, organization of videos into a taxonomy becomes more important. While an effective search engine can be used to find specific content, casual browsing by temporal or subject proximity is a task that cannot be entirely replaced by search engines.

Initially, the taxonomy can be defined by the video archivist. In our video library, the taxonomy hierarchy is defined by a tree with a depth of one, with a preference for an equal number of videos in each leaf node. For example, more than 400 videos in our library are taken from the course E1102 over a period of seven years, with as many as 130 videos in one year. Using our preference, we have identified eight sub-categories for E1102, which represent a reasonable granularity in grouping videos (see Figure 6.5).

We introduce a 3D category browser (see Figure 6.6), which emphasizes recurrences of categories over time, while placing similar categories in spatial proximity. We model the two parameters into a horizontally oriented cylinder, whose length represents linear time and circular surface area represents a continuous space (mathematically, this is the space $S^1$) over which category labels can be freely positioned. The cylinder is divided lengthwise into sections, each of which represents one year, although this choice of temporal granularity is arbitrary; semesters may be better. Category labels relevant for a given year are spaced out on cylinder’s cross section. In our implementation, we define the similarity between categories in the tree manually; however, it is not infeasible to generate the similarity matrix automatically from the similarities of existing transcripts of the videos, or from other comparable material.
Figure 6.5. Taxonomy for Category "E1102".

![Taxonomy Diagram for E1102](image1)

Figure 6.6. 3D Cylinder Taxonomy Browser. The cylinder's length (horizontal axis) represents time and is divided into sections, one per year. Category labels relevant for each year are spaced out on each section’s circumference. Temporally recurring categories are linked via curves on the cylinder’s surface. The 3D object is navigated by pressing and moving the mouse – horizontally to pan time and vertically to rotate the cylinder. Categories are selected by pressing on the labels.

![3D Cylinder Taxonomy Browser](image2)
The 3D cylinder UI was systematically derived from stacking multiple 2D planar graphs of categories within one time period (Figure 6.7). Nodes (categories) arrange freely under Multidimensional Scaling in the 2D Cartesian space to form the best configuration underlying pair-wise similarities. However, once mapped into the 3D space with the addition of the temporal dimension, we have decided to limit this configuration to the circumference of a circle (= surface area of cylinder). This restriction makes it
easier to navigate the 3D space without explicitly modeling depth as a dimension for category labels.

Navigation of the 3D cylinder requires pressing the mouse button and moving the mouse – horizontal movement changes the time period and vertical movement rotates the cylinder’s surface area. Mouse pointer distance from the originally clicked position defines the speed at which the cylinder is panned and rotated. Category labels on the cylinder’s visible area (front) are emphasized by color and can be selected, while labels on the invisible surface (rear) are colored in gray and cannot be selected. Categories with temporal recurrence are linked by curves on the cylinder’s surface.

Navigation between categories and their sub-categories retains the entire timeline to preserve temporal context. For example, selecting the category “E1102” for the year “2004” yields a new browser for sub-categories of E1102 centered on the year 2004, while also including sub-categories for all preceding and succeeding years relevant to E1102.

The spatial layout of labels within and across cylinder sections is determined by a distance minimization algorithm, whose error is computed from intra-section distances and weighted inter-section distances. We implement distance minimization in the $R^1S^1$ space using a heuristic approach since standard Multidimensional Scaling algorithms do not apply. Intra-section distances are computed over all pairs of labels, yielding a preferred localized layout. Weighted inter-section distances are computed only between recurring labels of a section and its three preceding sections, giving less weight to more
Figure 6.8. Distance Minimization between Nodes. Intra- and inter-section distances are computed between selected pairs of nodes. However, our approach does not require numerical distances between all pairs (e.g. Topic B – Topic D).

distant ones. This measure ensures that sections are well lined up over recurring labels. A second inter-section distance is computed between all non-shared labels between two consecutive sections, ensuring that dissimilar labels between two sections do not align to create a false sense of similarity.

This distance minimization problem cannot be formalized as a closed form linear algebra calculation such as the one introduced in [Pless and Simon, 2001], because it cannot guarantee numerical values for all distances between pairs of nodes in the distance matrix. Specifically, when a topic is introduced in section $N$ without prior use, it has an unidentified distance to all topics in section $N-2$ that do not recur (see Figure 6.8). The effect of unspecified distances between two nodes provides flexibility in label layouts of future sections. This is particularly helpful when categories no longer recur after a certain
point in time and they no longer require a position in the spatial layout. However, there is no way to represent these “don’t care” values in standard linear algebra. Instead we use a type of greedy algorithm with iterative refinement steps.

We have evaluated users’ perception of and interaction with the 3D cylinder representation for taxonomies in a user study with 74 students. Participants were given five category selection tasks, each requiring the selection of a main and a sub-category, e.g. “Select the category for the earliest final presentation videos in the course E1102”. On average, each successive task was completed in less time (56.5, 38.8, 33.5, 27.6, 20.6 sec) and successful task completion jumped from 77% to 96% after the first task and remained constant thereafter. The numerical results suggest a short learning curve with an almost linear time decrease after the first task. Additional user studies with larger sets of tasks would be required to find the point at which the learning curve finally flattens out. Survey results about ease of use and overall preference were mixed and bi-modal. About half of the participants found the novel UI very interesting and fun, while the other half found it useless and difficult. Ironically, all these students in the course were taught 3D modeling tools and concepts in CAD and Maya for three months before the user study took place.

The 3D cylinder taxonomy browser is a purely experimental interface, which combines temporal and spatial information. Additional work is required to compare it to more traditional interfaces, e.g. trees, lists, 2D graphs. The preliminary reaction by students is indicative of its popularity for some, suggesting that an alternative display may be necessary for the 2D-inclined.
Chapter 7

Characteristics of Video Browsing

The majority of video players in use provide controls for video playback, such as play, pause, stop, and location (fast forward/reverse) sliders, but do not implement summarization cues for content browsing and searching. In the absence of such cues, the search for content within and across videos becomes a cumbersome task which is reduced to linear view-and-skip-content operations. As the media database grows, the time and effort for content retrieval grows proportionally, and reaches its maximum when desired content occurs at random. We explored random content retrieval in standard video players, and also present results to compare the different interaction patterns between such standard players and index-cue driven browsers, such as VAST MM.

7.1 Index-Cue Browsing: Order of Magnitude Better

To better understand user interaction with standard video players and to provide a baseline measure for comparison to index-cue driven browsers, we established a user
study with 137 participants, 79 of who used a standard video player, and 58 of who used an index cue-driven video browser. All participants were given the same nine search tasks in randomized order per participant, and the same video dataset containing 204 student presentation videos with altogether 170 hours of audio-visual material (about 18 million video frames). Students were given 30 minutes to complete as many of the tasks as possible.

The standard video player (see Figure 7.1) includes the features play, pause, stop, as well as a location slider to quickly move to any position in the video. Additionally, the location slider provides a visual fast-forward view of keyframes as the knob is moved to another position. Using this control, the user is able to view the entire video in a matter of seconds. For practical implementation, keyframes are cached during video loading to ensure seamless browsing of this visual data; while not available in many standard video players, this feature was implemented to simulate the state-of-the-art in video players. The video player is embedded in an application that lists the entire video collection by titles, e.g. “Final Presentation, Section 1, Fall 2004, Tape 2”, etc. Without search features, the user must randomly select and browse videos in search for a task’s matching content. We thereby simulate a situation in which the user has access to a directory of videos without content-defining cues.

The index-cue driven video browser features searchable and browsable indices in addition to the features of the standard video player (see Figure 7.2). Automatically transcribed text is used as a searchable medium, while browsable visual cues, including
Figure 7.1. Video Browser with a List of Videos and a Standard Video Player with play/pause/stop buttons and a location slider. The player is augmented with keyframes for fast visual skimming.

Figure 7.2. Index-Cue Driven Browser featuring a list of text-searchable videos, video player, and browsable video summaries with audio, visual, and textual cues. Users can interactively change the amount and specificity of information in the video summary via various sliders (bottom).
Table 7.1. Distribution of search tasks and their matching video content. The entire video library contains 611,388 seconds of presentation video material collected from the same course over several years. Search tasks were designed to measure search for rare and common content.

<table>
<thead>
<tr>
<th>Type of Search</th>
<th>Search task “Find video content for presentations of or relating to …”</th>
<th>Time (sec) of Video Content</th>
<th>Percentage of Library Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specific Presentation</td>
<td>Musical Device for People with Cerebral Palsy</td>
<td>1,860</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>Coogan's Restaurant - Food Waste to Energy</td>
<td>1,860</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>CU Study Away - A Website for the Study Abroad Program</td>
<td>2,160</td>
<td>0.4</td>
</tr>
<tr>
<td>Specific Client</td>
<td>The MTA (Metropolitan Transportation Authority)</td>
<td>3,600</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>The 125th Street BID</td>
<td>7,200</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>Dr. Gil Lopez of CMSP Math Scales</td>
<td>14,400</td>
<td>2.4</td>
</tr>
<tr>
<td>Project Category</td>
<td>Information Technology (Web sites, Database projects, Handheld devices, etc.)</td>
<td>106,200</td>
<td>17.4</td>
</tr>
<tr>
<td></td>
<td>Architectural Design (e.g. Lab space design, Office design, etc.)</td>
<td>117,000</td>
<td>19.1</td>
</tr>
<tr>
<td></td>
<td>Disabilities</td>
<td>223,200</td>
<td>36.5</td>
</tr>
</tbody>
</table>

thumbnail images, speaker segments, and filtered keywords and phrases are displayed alongside the video player. The index-cue driven video browser is embedded in an application similar to the standard video player, which lists the entire video collection by title. A search tool can be used to locate videos with specific textual content.

Search tasks are designed to span a wide spectrum of information contained in the student presentation videos, in order to provide comprehensive search criteria for rare versus common content, as outlined in Table 7.1. For example, to find the presentation of a specific one-semester student project is considered a rare search – out of 170 hours of video, only 30 minutes of video count towards the correct answer. However, a search task
to find a specific project client, who has supervised multiple projects over a series of semester, can be answered by any one of multiple instances of video content. Finally, to search for a specific category of project can be answered by as much as 36.5% of all video content, due to the breadth of specific project categories.

7.2 Rigidity of Browsing “Styles”

Search and retrieval using standard video players without any index cues is expectedly inefficient. Students applied a variety of strategies to perform the search tasks, given the plain list of videos with only titles hinting at the semester in which they were shot. Strategies included random selection of videos; selection of only midterm or only final presentation videos, since project presentation content is similar in both of these course milestones; skipping entire years of presentations because they exhibited only one or two task-unrelated categories.

The video player provided to students offered two methods of browsing video content. The “audio-visual” method reflects a sequence of video clip playbacks and temporal skips. The “visual” method is based solely on fast-forward keyframe skimming, in which little or no video, and no audio, playback take place. We observe that students generally adhere to only one of these approaches during the user study; 50% of students prefer audio-visual browsing, while the other half prefers visual browsing. However, the number of attempted and correctly completed tasks is significantly higher, and task duration is lower, for audio-visual browsing (see Table 7.2). The advantage of audio-visual browsing is likely due to additional cues from audio, whereas strictly visual presentation material is sometimes too terse of a representation.
Table 7.2. Audio-visual versus visual browsing. Their preference among users is equally distributed. However, audio-visual browsing tends to outperform visual-only browsing in search tasks.

<table>
<thead>
<tr>
<th></th>
<th>Users</th>
<th>Attempted Tasks</th>
<th>Task Time</th>
<th>Correct Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio-Visual Browsing</td>
<td>39</td>
<td>97</td>
<td>531 sec</td>
<td>36</td>
</tr>
<tr>
<td>Visual Browsing</td>
<td>40</td>
<td>56</td>
<td>846 sec</td>
<td>15</td>
</tr>
</tbody>
</table>

Using the data from user studies on standard video players, we are able to qualify the user interaction during search and retrieval tasks. Audio-visual browsing can be described as the repetitive event of listening and viewing video clips of some duration then skipping a certain amount of time to the next video clip. We found no significant correlation between these two steps, that is, the amount of video viewed does not predict amount of video skipped. We summarize the unrelated distributions of length of video clip viewed and duration of video skipped in Figures 7.3 and Figures 7.4 (blue lines). Most users view between one and eight seconds of video, then skip forward between 30 and 300 seconds or backwards between 30 and 60 seconds.

Visual browsing can be described as the repetitive event of viewing a keyframe then visually skimming a set of them at higher speed to the next keyframe of interest. Measuring what keyframes the user viewed and which were skimmed quickly is not well defined. Depending on the user’s cognitive abilities, the content of a keyframe can be visually processed in less than 100 milliseconds. According to the empirical evaluation reported in [Yu et. al, 2007], a person can read as many as 600 words per minute in a flashcard setting, which, on average, is one word every 100 ms. Similar results were
Figures 7.3. Distributions for Duration of video clips viewed during Audio-Visual Browsing (blue) and keyframes viewed during Visual-only browsing (red). While their temporal scales are different by a factor of 10, distributions show a similar long tail.

Figures 7.4. Distributions for Time Skipped between video clips during Audio-Visual Browsing (blue) and time skipped between keyframes during Visual-only browsing (red). The distribution is similar for both approaches, with the temporal scale varying by a factor of two.

found in [Thorpe et. al., 1996] for recognizing objects in scenes, e.g. “animals”. Subjects were able to identify scene objects within 20 ms and process the information within an additional 150 ms when previously unseen photographs were flashed before them. Based on these evaluations, we consider a keyframe skipped if it was viewed in less than 200 ms. This duration is enough to read two words from a keyframe if it contains a presentation slide. Figures 7.3 and Figures 7.4 (red lines) present the distributions for duration of keyframe viewed and temporal distance between keyframes skipped. Most users view a keyframe in less than 0.5 seconds before skipping forward to the next keyframe between 50 and 200 seconds or to a previous keyframe up to 50 seconds backwards.
Figure 7.5. Task Time: Index-Cue Driven Browser vs. Video Player: Required time to complete a search task depends highly on available cues. With a standard video player, search time varies inversely with matching content, while an index-cue driven browser offers near constant search time. The x-axis is logarithmically scaled.

We observe that the temporal distance skipped between video content is similar between the two browsing approaches. However, while the duration during which content is viewed differs between audio-visual and visual-only browsing; these results indicate that visual-only browsing occurs at much higher speeds. Nevertheless, it is insensitive to potentially important content only available in audio material, leading to lower scores.

7.3 Conclusions

Without searchable and browsable cues in video browsers, the time required to locate specific content grows inversely to the amount of video content that matches the query. A video browser with the salient cues, however, should show a near-constant
Standard video player

Search for rare content

Search for less-rare content

Search for common content

Index-cue driven browser

Search for rare content

Search for less-rare content

Search for common content

Figures 7.6a-c. Search Task Results for Standard Video Player. With the exception of finding common content, video players are ineffective for search and retrieval. Top shaded (red) portion of each bar indicates unsuccessful, bottom shaded (blue) successful searches.

Figures 7.7a-c. Search Task Results from an Index-Cue Driven Browser. Index cues are useful for finding video content regardless of number of matching video clips. Top shaded (red) portion of each bar indicates unsuccessful, bottom shaded (blue) successful searches.
amount of time required to locate content of varying recurrence, a phenomenon which we
can document (see Figure 7.5).

As anticipated, a standard video player is ineffective for those search tasks which
seek very unusual matching content. Figures 7.6(a) and (b) demonstrate that of all
attempted tasks, the majority of them failed. When completed successfully, the average
time required exceeded 10 minutes. However, a search task aimed at locating common
information with more than 15% of the corpus satisfying this task was typically
completed successfully at reasonable speed with the same standard video player (see
Figures 7.6(c)).

Most significantly, when an index-cue driven browser was available to perform
the same search tasks, completion and required time remain comparable throughout,
regardless of difficulty (see Figures 7.7(a-c)). Overall completion rates exceed 70% as
compared to 33%, and average required time of 110 seconds is also significantly lower
than 646 seconds for standard video players.
Chapter 8

Conclusions and Future Work

In this thesis, we have introduced approaches for indexing unstructured videos via multi-modal cues, methods of visualizing this information in a novel multimedia (VAST MM) browser, and evaluations of extensive user studies over a three-year period validating the utility and necessity of such tools.

8.1 Contributions

We have introduced effective methods of extracting and indexing multi-modal cues from presentation videos, a genre which has not been investigated in depth prior to this work. Meaningful cues identified and evaluated include scene segmentation of unedited video and extraction of representative keyframes, speaker segmentation and extraction of representative speaker faces, speaker clustering to determine speaker recurrences in a video, extraction of text from speech and filtering of key phrases, and higher level structure detection based on presentation changes in a video containing more
than one presentation. All of these measures were evaluated for their effectiveness in a video browser, and with a few exceptions all of them have been successfully automated.

We have introduced a novel multimedia browser, the first of its kind, for unstructured videos. The browser makes available browsable cues, including keyframes, speaker segments, face indices, and filtered keyphrases in addition to a standard video player augmented with keyframes for skimming. The interface provides tools which can be used to interactively vary the amount of information displayed and to produce summaries that are more accurately tuned to a user’s level of comfort.

We have performed frequent user studies over a three-year period, testing aspects of content indices and the user interface, and improving them based on feedback and evaluation of search and retrieval tasks. More than 1,000 unique students have used the VAST MM system for review of presentations and study for exams. In our evaluation, we have made several noteworthy observations. Structural cues, such as a zoomable hierarchy of various segmentations, significantly improve browsing of unstructured videos by reducing the time required to locate content. For targeted search and retrieval of video information, the presence of the video itself is counter-productive – using index cues alone significantly decreases the time required to locate content without sacrificing accuracy. Finally, the availability of multimedia content through a browser such as VAST MM is an effective resource for exam preparation. In a related user study, we have determined that users who make effective use of this resource experience an improvement in exam scores.
In our early work, we have investigated novel user interfaces and indices for lecture video content. Through classification of keyframes and clustering of lecture content in the form of handwritten material into topics, we were able to organize this otherwise serial information into semantic units. Through user studies performed with a user interface disseminating these indices, we validated their utility for browsing of lecture video content.

Our analysis of index cues and their utility for content retrieval in the scope of this work focused on instructional and presentation videos. As part of this research, we also briefly investigated a third genre of unstructured videos, those of team interactions and meetings, to validate our indexing approaches on them. Team interaction and meeting videos tend to be captured in very casual settings and their content varies widely, ranging from semi-formal meeting-room meetings to very informal dorm-room style gatherings, from almost still team interaction in a computer lab to vividly active footage of teams making field visits to clients in an outside setting. These videos exhibit greater audio-visual variability than those we have identified in lecture and presentation videos: semi-formal meetings oftentimes contain less visual activity than presentations, whereas field visits sometimes include long periods of footage captured while the operator is walking on streets or in parks. Audio quality for videos in this genre is generally poor – rarely are external microphones used to supplement the low quality camera microphone. Extracting text from speech is therefore extremely challenging, and external filters consequently failed to emphasize key terms. Additional research into methods of indexing and content retrieval from videos with such highly varying audio-visual
qualities is required and necessary because they most closely parallel home videos, a genre with a large audience.

8.2 Future Work

With growing video archives and increasing use of this medium, whether on the desktop or in the mobile environment, a vast amount of research still lies ahead. In this thesis, we have identified several approaches for building video indices, some of which were implemented and tested over several semesters, others of which were prototyped for single studies, with very promising outcomes. We now identify several specific future directions which we would like to address:

*Structure across courses:* Course lectures tend to be recorded every term in which they are offered – this consequently leads to several versions of the same course taught by different instructors. Depending on teaching style, trends, and preference, different instructors may place different emphasis on various topics. It would be feasible through text analysis to correlate teaching topics between several versions of the same course over time. Such analysis would be helpful in identifying trends, help faculty in evaluating course topics, and, if available to students, provide alternative resources for the same material.

*Efficient lecture content clustering:* We have demonstrated that visual content-based clustering for lecture videos is a powerful approach for content organization. Our comparison metric between two images, however, computationally complex – it explicitly models the zoom parameter of the camera by using an empirically derived set of scale factors. And, while we did not find it necessary to explicitly model perspective
transforms due to the limited viewpoints of the camera, we cannot generally dismiss this parameter. To improve the efficiency and effectiveness of pair-wise image comparisons, methods for parameter determination for perspective and scale must be applied to images. Alternatively, scale-invariant approaches for content comparison can also be applied, for example by detecting primitive shapes via Hough transforms [Ballard, 1981].

**Clustering of electronic slides:** Our early work on lecture video content clustering focused on designing filters and comparison methods for blackboard and hand-drawn sheet keyframes. While this work is still applicable, we have observed a significant increase in the use of electronic slides as the dominant presentation medium in some classes. This shift in teaching style must be accommodated by similar content clustering methods to retain the favorable reduction in visual material demonstrated in our user studies. Approaches for clustering electronic slides are not based on visual comparisons, but instead on textual features, whether derived from optical character recognition or speech transcripts.

**Search results with examples:** We have identified a potential gap in the present information retrieval flow when search is performed. Search results for textual queries are provided as a list of matching videos in ranked order. Any selected video is then retrieved and displayed in the video content browser in its entirety. It will be beneficial for the user to view some example video material in a step before the detailed video information. This information will include visual cues from keyframes, text cues from filtered text, and audio cues. Web-based text search engines provide a similar in-between step by highlighting search terms and displaying adjacent text in the search result page.
Transcript segmentation: Clustering of text into topics for lecture videos finds application in other video domains as well. Raw student presentation videos oftentimes feature several distinct presentations in one video. Text clustering will be helpful in identifying thematic breaks, which are likely coincidental with breaks between presentations. When such high-level structure in video is identifiable, more effective tables of content and semantic summaries of video segments can be generated.

Annotation threads: We have introduced location sensitive annotations and bookmarks. Annotations are intended as means for publically visible discussions among users. In their present form, however, they exist as individual text call-outs, and would benefit from thread-like discussion behavior. The extent to which annotations develop social networks is unclear.

Popularity: We can leverage information from our detailed interaction logs to establish a more implicit social networking approach to emphasizing information in the browser. Through analysis, we can determine popularity metrics for categories, videos, and even fine-grained information such as specific keyframes and text cues. Along with automatically extracted cues, various popularity measures can be included in the browsing interface. With its present modular track-like design, users can easily enable/disable these cues to generate a preferred view of the video information.

Phoneme-based text search: Transcripts from automatic speech recognition and approaches for filtering keywords and key phrases have proven to generate invaluable cross-video text indices for lecture and student presentation videos. This process fails, however, on meeting/interaction videos of lower audio quality. In general, we observe
that undirected microphones, such as camera built-in ones are insufficient for capturing speech with an adequately low level of noise, as required by ASR systems. An alternative to performing semantic-level speech recognition may be better suited for speech-level search, such as a phonetic-based approach, in which detected phonemes are not represented by best-matching words. While this approach is computationally more costly due to the additional complexity of matching phonemes, bi-grams, and tri-grams of varying probability, it makes available matches which are otherwise lost on the semantic level.

*Optical character recognition:* Automatic speech recognition is a reasonable source for text cues. However, written text in videos, whether from slides in lecture and presentation videos, or other artifacts in meeting/interaction videos will prove invaluable as an additional source for text cues. In preliminary experiments, we have determined that conventional OCR systems have substantial difficulties extracting coherent text from candid video shots due to the lack of visual quality (e.g. contrast and perspective transforms). Even though our experiments with four OCR platforms focused on printed and not handwritten text, we have not had reasonable success. Research into new or modified approaches is necessary to provide a text-from-image process.

*Voice thumbnails:* In addition to visual face cues in the form of thumbnails, voice prints in the form of audio thumbnails will enhance browsing for individual speakers. A short duration audio print of this kind will contain characteristic speech for a speaker, which can be identified through MFCC vectors that are closest to the center of a speaker’s MFCC distribution.
Visual concept detection: The current implementation of video search relies heavily on text from ASR, yet videos contain information far beyond speech. Research in the area of news videos has shown the viability of visual concept detection for search [Natsev et. al., 2004; Kender and Naphade, 2005]. A training set of images is first annotated manually with a limited set of concepts, which are then mapped to characteristic visual features in a higher-dimensional space, such as color histograms, edges, texture, etc. Feature selection in this space is typically performed with machine learning methods, such as support vector machines (SVM) and Gaussian mixture models (GMM). The resulting concept models are then applied to previously unclassified images to determine their matching probabilities. For unstructured videos, a taxonomy of concepts must first be designed and a training set of images must be annotated. We have already added a visual concept annotation tool to the VAST MM browser for this purpose. Similar to semantic search in the visual space of news videos [Haubold et. al., 2006], text search for unstructured videos can be fused with the visual concept space for improved retrieval performance.

Cross-video information threads: With the improvement of index cues for individual videos, research in finding threads of information across videos becomes possible. Current technology provides such threads by means of text searches. However, multi-modal information from videos introduces new dimensions, such as recurring actors, which can be measured visually by faces, or audibly by voice and speech patterns. The combination of many cues and the contextually meaningful representation of cross-video cues in one browser are true multimedia challenges.
Cross-video speaker clustering: Face indices are effective in representing speakers and their recurrences in a video. Models of this kind are highly useful for non-linear browsing of video content. While underlying work exists for speaker clustering and face detection, combining these research topics is by no means considered completed, in particular for the domain of candidly captured videos. Additional research into face detection and fusion of speaker clustering using audio and visual cues is necessary to create completely automated approaches for generating face indices.

Video-to-text maps: We have shown that reasonably accurate mappings can be established between transcripts and external corpora for lecture videos. Course textbooks are, however, very structured and we anticipate that a strong mapping exists. Further fundamental research is required in determining contextual links between video contents and large external data corpora, such as a library of books, or possibly the WWW. Mappings between text documents and videos can be used two-fold: for informational purposes, videos can suggest external material of similar content, while external documents can provide structural clues about a video. The latter is of particular interest for unstructured videos as we have shown in user studies. When presentation videos were indexed in a textbook chapter/section format, users were able to find information much faster compared to users who had no structural cues.

Other genres: Finally, work in this research area is relevant to domains beyond the university environment. Determining cues relevant to domains such as home videos, uncut news videos, etc. will benefit a wider audience. We anticipate that home videos are closely related to meeting/interaction videos in our database in their audio-visual
qualities. Visual information ranges from very steady and still shots (e.g. shots of a building or a group of people at a BBQ) to wildly moving ones (e.g. shots out of a moving car window or video recorded while walking). With the occasional exception of speech from the camera operator, any audio recorded from actors at a distance is oftentimes heavily affected by noise. We have determined that our visual segmentation and keyframe extraction methods are suitable for these videos; however, text-based indexing methods fail.
### Appendix A

Stop word list used for text filtering.

<table>
<thead>
<tr>
<th>a</th>
<th>anyway</th>
<th>can</th>
<th>et</th>
<th>further</th>
</tr>
</thead>
<tbody>
<tr>
<td>about</td>
<td>anywhere</td>
<td>cannot</td>
<td>etc</td>
<td>furthermore</td>
</tr>
<tr>
<td>above</td>
<td>apart</td>
<td>canst</td>
<td>even</td>
<td>furthest</td>
</tr>
<tr>
<td>according</td>
<td>are</td>
<td>certain</td>
<td>ever</td>
<td>g</td>
</tr>
<tr>
<td>across</td>
<td>around</td>
<td>cf</td>
<td>every</td>
<td>get</td>
</tr>
<tr>
<td>after</td>
<td>as</td>
<td>choose</td>
<td>everybody</td>
<td>go</td>
</tr>
<tr>
<td>afterwards</td>
<td>at</td>
<td>contrariwise</td>
<td>everyone</td>
<td>h</td>
</tr>
<tr>
<td>again</td>
<td>av</td>
<td>cos</td>
<td>everything</td>
<td>had</td>
</tr>
<tr>
<td>against</td>
<td>b</td>
<td>could</td>
<td>everywhere</td>
<td>halves</td>
</tr>
<tr>
<td>albeit</td>
<td>be</td>
<td>cu</td>
<td>except</td>
<td>hardly</td>
</tr>
<tr>
<td>all</td>
<td>became</td>
<td>d</td>
<td>excepted</td>
<td>has</td>
</tr>
<tr>
<td>almost</td>
<td>because</td>
<td>day</td>
<td>excepting</td>
<td>hast</td>
</tr>
<tr>
<td>alone</td>
<td>become</td>
<td>did</td>
<td>exception</td>
<td>hath</td>
</tr>
<tr>
<td>along</td>
<td>becomes</td>
<td>do</td>
<td>exclude</td>
<td>have</td>
</tr>
<tr>
<td>already</td>
<td>becoming</td>
<td>does</td>
<td>excluding</td>
<td>he</td>
</tr>
<tr>
<td>also</td>
<td>been</td>
<td>doesn</td>
<td>exclusive</td>
<td>hence</td>
</tr>
<tr>
<td>although</td>
<td>before</td>
<td>doing</td>
<td>f</td>
<td>henceforth</td>
</tr>
<tr>
<td>always</td>
<td>beforehand</td>
<td>doing</td>
<td>far</td>
<td>her</td>
</tr>
<tr>
<td>among</td>
<td>behind</td>
<td>dost</td>
<td>farther</td>
<td>here</td>
</tr>
<tr>
<td>amongst</td>
<td>being</td>
<td>doth</td>
<td>farthest</td>
<td>hereabouts</td>
</tr>
<tr>
<td>am</td>
<td>below</td>
<td>double</td>
<td>few</td>
<td>hereafter</td>
</tr>
<tr>
<td>an</td>
<td>beside</td>
<td>down</td>
<td>ff</td>
<td>hereby</td>
</tr>
<tr>
<td>and</td>
<td>besides</td>
<td>dual</td>
<td>first</td>
<td>herein</td>
</tr>
<tr>
<td>another</td>
<td>between</td>
<td>during</td>
<td>for</td>
<td>hereto</td>
</tr>
<tr>
<td>any</td>
<td>beyond</td>
<td>e</td>
<td>formerly</td>
<td>hereupon</td>
</tr>
<tr>
<td>anybody</td>
<td>both</td>
<td>each</td>
<td>forth</td>
<td>hers</td>
</tr>
<tr>
<td>anyhow</td>
<td>but</td>
<td>either</td>
<td>forward</td>
<td>herself</td>
</tr>
<tr>
<td>anyone</td>
<td>by</td>
<td>else</td>
<td>from</td>
<td>him</td>
</tr>
<tr>
<td>anything</td>
<td>c</td>
<td>elsewhere</td>
<td>front</td>
<td>himself</td>
</tr>
<tr>
<td>hindmost</td>
<td>L</td>
<td>nevertheless</td>
<td>outside</td>
<td>should</td>
</tr>
<tr>
<td>----------</td>
<td>---</td>
<td>--------------</td>
<td>---------</td>
<td>--------</td>
</tr>
<tr>
<td>his</td>
<td>last</td>
<td>next</td>
<td>over</td>
<td>shown</td>
</tr>
<tr>
<td>hither</td>
<td>latter</td>
<td>no</td>
<td>own</td>
<td>sideways</td>
</tr>
<tr>
<td>hitherto</td>
<td>latterly</td>
<td>nobody</td>
<td>p</td>
<td>since</td>
</tr>
<tr>
<td>how</td>
<td>less</td>
<td>none</td>
<td>per</td>
<td>slept</td>
</tr>
<tr>
<td>however</td>
<td>lest</td>
<td>nonetheless</td>
<td>perhaps</td>
<td>slew</td>
</tr>
<tr>
<td>howsoever</td>
<td>let</td>
<td>noone</td>
<td>plenty</td>
<td>slung</td>
</tr>
<tr>
<td>i</td>
<td>like</td>
<td>nope</td>
<td>provide</td>
<td>slunk</td>
</tr>
<tr>
<td>ie</td>
<td>little</td>
<td>nor</td>
<td>q</td>
<td>smote</td>
</tr>
<tr>
<td>if</td>
<td>ltd</td>
<td>not</td>
<td>quite</td>
<td>so</td>
</tr>
<tr>
<td>in</td>
<td>m</td>
<td>nothing</td>
<td>r</td>
<td>some</td>
</tr>
<tr>
<td>inasmuch</td>
<td>many</td>
<td>notwithstanding</td>
<td>rather</td>
<td>somebody</td>
</tr>
<tr>
<td>inc</td>
<td>may</td>
<td>now</td>
<td>really</td>
<td>somehow</td>
</tr>
<tr>
<td>include</td>
<td>maybe</td>
<td>nowadays</td>
<td>round</td>
<td>somehowhow</td>
</tr>
<tr>
<td>included</td>
<td>me</td>
<td>nowhere</td>
<td>s</td>
<td>someone</td>
</tr>
<tr>
<td>including</td>
<td>meantime</td>
<td>o</td>
<td>sake</td>
<td>something</td>
</tr>
<tr>
<td>indeed</td>
<td>meanwhile</td>
<td>of</td>
<td>same</td>
<td>sometime</td>
</tr>
<tr>
<td>indoors</td>
<td>might</td>
<td>off</td>
<td>sang</td>
<td>sometimes</td>
</tr>
<tr>
<td>inside</td>
<td>moreover</td>
<td>often</td>
<td>save</td>
<td>somewhat</td>
</tr>
<tr>
<td>insomuch</td>
<td>most</td>
<td>ok</td>
<td>saw</td>
<td>somewhere</td>
</tr>
<tr>
<td>instead</td>
<td>mostly</td>
<td>on</td>
<td>see</td>
<td>speak</td>
</tr>
<tr>
<td>into</td>
<td>more</td>
<td>once</td>
<td>seeing</td>
<td>spat</td>
</tr>
<tr>
<td>inward</td>
<td>mr</td>
<td>one</td>
<td>seem</td>
<td>spoken</td>
</tr>
<tr>
<td>inwards</td>
<td>mrs</td>
<td>only</td>
<td>seem</td>
<td>sprang</td>
</tr>
<tr>
<td>is</td>
<td>ms</td>
<td>onto</td>
<td>seemed</td>
<td>sprung</td>
</tr>
<tr>
<td>it</td>
<td>much</td>
<td>or</td>
<td>seeming</td>
<td>stave</td>
</tr>
<tr>
<td>its</td>
<td>must</td>
<td>other</td>
<td>seems</td>
<td>staves</td>
</tr>
<tr>
<td>itself</td>
<td>my</td>
<td>others</td>
<td>seen</td>
<td>still</td>
</tr>
<tr>
<td>j</td>
<td>myself</td>
<td>otherwise</td>
<td>seldom</td>
<td>such</td>
</tr>
<tr>
<td>just</td>
<td>n</td>
<td>ought</td>
<td>selves</td>
<td>supposing</td>
</tr>
<tr>
<td>k</td>
<td>namely</td>
<td>our</td>
<td>sent</td>
<td>t</td>
</tr>
<tr>
<td>kind</td>
<td>need</td>
<td>ours</td>
<td>several</td>
<td>than</td>
</tr>
<tr>
<td>kg</td>
<td>neither</td>
<td>ourselves</td>
<td>shalt</td>
<td>that</td>
</tr>
<tr>
<td>km</td>
<td>never</td>
<td>out</td>
<td>she</td>
<td>the</td>
</tr>
<tr>
<td>thee</td>
<td>throughout</td>
<td>using</td>
<td>wherefrom</td>
<td>whose</td>
</tr>
<tr>
<td>---</td>
<td>---------------------</td>
<td>----------------</td>
<td>-----------------</td>
<td>-----------</td>
</tr>
<tr>
<td>their</td>
<td>thru</td>
<td>v</td>
<td>wherein</td>
<td>whosoever</td>
</tr>
<tr>
<td>them</td>
<td>thus</td>
<td>very</td>
<td>whereinto</td>
<td>why</td>
</tr>
<tr>
<td>themselves</td>
<td>thy</td>
<td>via</td>
<td>whereof</td>
<td>will</td>
</tr>
<tr>
<td>then</td>
<td>thyself</td>
<td>vs</td>
<td>whereon</td>
<td>wilt</td>
</tr>
<tr>
<td>thence</td>
<td>till</td>
<td>w</td>
<td>wheresoever</td>
<td>with</td>
</tr>
<tr>
<td>thenceforth</td>
<td>to</td>
<td>want</td>
<td>whereto</td>
<td>within</td>
</tr>
<tr>
<td>there</td>
<td>together</td>
<td>was</td>
<td>whereunto</td>
<td>without</td>
</tr>
<tr>
<td>thereabout</td>
<td>too</td>
<td>we</td>
<td>wherever</td>
<td>worse</td>
</tr>
<tr>
<td>thereabouts</td>
<td>toward</td>
<td>week</td>
<td>wherewith</td>
<td>worst</td>
</tr>
<tr>
<td>thereafter</td>
<td>towards</td>
<td>well</td>
<td>whether</td>
<td>would</td>
</tr>
<tr>
<td>thereby</td>
<td>u</td>
<td>were</td>
<td>whew</td>
<td>wow</td>
</tr>
<tr>
<td>therefore</td>
<td>ugh</td>
<td>what</td>
<td>which</td>
<td>y</td>
</tr>
<tr>
<td>therein</td>
<td>unable</td>
<td>whatever</td>
<td>whichever</td>
<td>ye</td>
</tr>
<tr>
<td>thereof</td>
<td>under</td>
<td>whatsoever</td>
<td>whichsoever</td>
<td>yet</td>
</tr>
<tr>
<td>thereon</td>
<td>underneath</td>
<td>when</td>
<td>while</td>
<td>year</td>
</tr>
<tr>
<td>thereto</td>
<td>unless</td>
<td>whence</td>
<td>whilst</td>
<td>yippee</td>
</tr>
<tr>
<td>thereupon</td>
<td>unlike</td>
<td>whenever</td>
<td>whither</td>
<td>you</td>
</tr>
<tr>
<td>these</td>
<td>until</td>
<td>whenever</td>
<td>who</td>
<td>your</td>
</tr>
<tr>
<td>they</td>
<td>up</td>
<td>whenssoever</td>
<td>whoa</td>
<td>yours</td>
</tr>
<tr>
<td>this</td>
<td>upon</td>
<td>where</td>
<td>whoever</td>
<td>yourself</td>
</tr>
<tr>
<td>those</td>
<td>upward</td>
<td>whereabouts</td>
<td>whole</td>
<td>yourselves</td>
</tr>
<tr>
<td>thou</td>
<td>upwards</td>
<td>whereafter</td>
<td>whom</td>
<td>yourselves</td>
</tr>
<tr>
<td>though</td>
<td>us</td>
<td>whereas</td>
<td>whomsoever</td>
<td>z</td>
</tr>
<tr>
<td>thrice</td>
<td>use</td>
<td>whereat</td>
<td>whomever</td>
<td></td>
</tr>
<tr>
<td>through</td>
<td>used</td>
<td>whereby</td>
<td>whomever</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>wherefore</td>
<td>whomever</td>
<td></td>
</tr>
</tbody>
</table>
## Appendix B

List of videos by source and genre used for the work in this thesis. The table lists number of videos and their total duration (in sec).

<table>
<thead>
<tr>
<th>Semester</th>
<th>Midterm Presentations</th>
<th>Final Presentations</th>
<th>Student Introductions</th>
<th>Team Meetings</th>
<th>Team Interactions</th>
<th>Student Interviews</th>
<th>Lectures</th>
<th>Events</th>
<th>Talks</th>
<th># Time (HR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall 1996</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1 1854</td>
</tr>
<tr>
<td>Fall 1999</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2002</td>
<td>7 17980</td>
<td>13 35398</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2003</td>
<td>5 15674</td>
<td>13 35524</td>
<td>5 11630</td>
<td>1 2556</td>
<td>5 10341</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summer 2003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2003</td>
<td>5 15897</td>
<td>7 29119</td>
<td>4 8374</td>
<td>3 9876</td>
<td>19 50253</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2004</td>
<td>9 21959</td>
<td>13 37255</td>
<td>5 8556</td>
<td>17 49186</td>
<td>25 67073</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summer 2004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2004</td>
<td>8 27124</td>
<td>9 32049</td>
<td></td>
<td>15 37947</td>
<td>32 100782</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2005</td>
<td>8 28386</td>
<td>12 36473</td>
<td></td>
<td>25 71283</td>
<td>38 122705</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summer 2005</td>
<td>2 6704</td>
<td>2 6819</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2005</td>
<td>9 29603</td>
<td>9 32654</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2006</td>
<td>8 25025</td>
<td>8 28822</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summer 2006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2006</td>
<td>8 23380</td>
<td>7 21322</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2007</td>
<td>8 22423</td>
<td>8 21165</td>
<td></td>
<td></td>
<td></td>
<td>12 39187</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summer 2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2007</td>
<td>8 23311</td>
<td>8 20321</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Semester</td>
<td>Midterm Presentations</td>
<td>Final Presentations</td>
<td>Student Introductions</td>
<td>Team Meetings</td>
<td>Team Interactions</td>
<td>Student Interviews</td>
<td>Lectures</td>
<td>Events</td>
<td>Talks</td>
<td># Time (HR)</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------</td>
<td>---------------------</td>
<td>-----------------------</td>
<td>---------------</td>
<td>------------------</td>
<td>--------------------</td>
<td>----------</td>
<td>--------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>#</td>
<td>Time</td>
<td>#</td>
<td>Time</td>
<td>#</td>
<td>Time</td>
<td>#</td>
<td>Time</td>
<td>#</td>
<td>Time</td>
</tr>
<tr>
<td>Videos - Columbia Video Network</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2003</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2007</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>82536</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>107662</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>235430</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>53</td>
<td>241199</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>253</td>
<td>1327296</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>50273</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>48</td>
<td>211189</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>474</td>
<td>626.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Videos - Columbia University Medical Center</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2008</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>51565</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>14.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Videos - Computer Science Department</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spring 2006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Summer 2006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2006</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>17738</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>5815</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4579</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>7.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>789</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>#</td>
<td>85</td>
<td>110</td>
<td>14</td>
<td>62</td>
<td>122</td>
<td>5</td>
<td>493</td>
<td>1</td>
<td>7</td>
<td>899</td>
</tr>
<tr>
<td>Time (Hours)</td>
<td>71.5</td>
<td>93.9</td>
<td>7.9</td>
<td>47.7</td>
<td>94.7</td>
<td>2.9</td>
<td>651.8</td>
<td>0.5</td>
<td>78</td>
<td>978.7</td>
</tr>
</tbody>
</table>
Appendix C

List of user studies, including dates, types of study, number of users, and breakdown of specially tested browser features.

<table>
<thead>
<tr>
<th>Course</th>
<th>Participants</th>
<th>User participation outlined in contingency tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1102</td>
<td>First-year course students</td>
<td>With video</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Without video</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monday</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;34,0,2&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;45,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;79,0,2&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tuesday</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;41,0,1&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;32,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;73,0,1&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;75,0,3&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;77,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;152,0,3&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>With video</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Without video</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;10,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;7,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;17,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td>Senior students (advisors)</td>
<td>Date: 4/29/2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>With video</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Without video</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;7,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td>External, adult audience (course</td>
<td>Date: 4/14/2005</td>
</tr>
<tr>
<td></td>
<td>unrelated)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>With video</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fall 2005</td>
<td></td>
<td>With video</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Without video</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monday</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;27,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;53,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;80,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tuesday</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;88,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;16,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;84,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;95,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;69,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;164,0,0&gt;</td>
</tr>
<tr>
<td>Spring 2006</td>
<td></td>
<td>Structured cues: keyframes + pres. seg.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Semi-structured cues: keyframes + text</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Semi-structured cues: video + pres. seg.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unstructured cues: video + text</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monday</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;18,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;20,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;22,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;18,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;78,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tuesday</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;40,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;38,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;39,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;40,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;157,0,0&gt;</td>
</tr>
<tr>
<td>Fall 2006</td>
<td></td>
<td>Face index: extreme close-up headshot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Face index: extreme close-up head and profile shot pair</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Face index: medium close-up headshot</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Face index: medium close-up head and profile shot pair</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Monday (in dorm)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;10,3,7&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;16,1,6&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;21,1,3&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;6,1,8&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;53,6,24&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tuesday (in class)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;20,0,2&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;20,0,1&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;14,0,2&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;25,0,0&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;80,0,5&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;30,3,9&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;36,1,7&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;35,1,5&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;32,1,8&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;133,6,29&gt;</td>
</tr>
<tr>
<td>Course</td>
<td>Participants</td>
<td>User participation outlined in contingency tables</td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
</tbody>
</table>
Monday <79,0,0>  
Tuesday <88,0,0>  
|        |              | April 2007:  
Monday <167,0,0>  |
Monday <79,0,0>  
Tuesday <88,0,0>  
|        |              | April 2007:  
Monday <167,0,0>  |
| W4771  | Senior undergraduate and graduate students | Dates: 5/2/2007 - 5/10/2007  
First trial with generic guest logins. Here we count unique user sessions:  
<52,unknown>  |
Monday <74,0,0>  
Tuesday <86,0,0>  
|        |              | April 2007:  
Monday <160,0,0>  |
Monday (in dorm) <74,0,0>  
Tuesday (in class) <68,1,17>  
|        |              | April 2007:  
Monday (in dorm) <142,1,17>  |
| W4115  | Senior undergraduate and graduate students | Dates: 11/30/2007 - 12/6/2007  
Participants <27,42>  |
| W4824  | Senior undergraduate and graduate students | Dates: 12/6/2007 - 12/18/2007  
Participants <30,64>  |

**Key:**
- **Targeted task user study**
- **General usage for project background research in course**
- **Casual final exam study**

3-tuples denote user study participation (unique users):  
< #users completed study, # users did not complete, # users did not participate >  
2-tuples denote usage pattern for casual use (unique users):  
< #users participated, # users did not participate >
Bibliography


