Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

ANNOTATION OF MULTIMEDIA LEARNING MATERIALS FOR SEMANTIC SEARCH

by
Sheetal Rajgure

Multimedia is the main source for online learning materials, such as videos, slides and textbooks, and its size is growing with the popularity of online programs offered by Universities and Massive Open Online Courses (MOOCs). The increasing amount of multimedia learning resources available online makes it very challenging to browse through the materials or find where a specific concept of interest is covered. To enable semantic search on the lecture materials, their content must be annotated and indexed. Manual annotation of learning materials such as videos is tedious and cannot be envisioned for the growing quantity of online materials. One of the most commonly used methods for learning video annotation is to index the video, based on the transcript obtained from translating the audio track of the video into text. Existing speech to text translators require extensive training especially for non-native English speakers and are known to have low accuracy.

This dissertation proposes to index the slides, based on the keywords. The keywords extracted from the textbook index and the presentation slides are the basis of the indexing scheme. Two types of lecture videos are generally used (i.e., classroom recording using a regular camera or slide presentation screen captures using specific software) and their quality varies widely. The screen capture videos, have generally a good quality and sometimes come with metadata. But often, metadata is not reliable and hence image processing techniques are used to segment the videos. Since the learning videos have a static background of slide, it is challenging to detect the shot boundaries. Comparative analysis of the state of the art techniques to determine best feature descriptors suitable for detecting transitions in a learning video

is presented in this dissertation. The videos are indexed with keywords obtained from slides and a correspondence is established by segmenting the video temporally using feature descriptors to match and align the video segments with the presentation slides converted into images. The classroom recordings using regular video cameras often have poor illumination with objects partially or totally occluded. For such videos, slide localization techniques based on segmentation and heuristics is presented to improve the accuracy of the transition detection.

A region prioritized ranking mechanism is proposed that integrates the keyword location in the presentation into the ranking of the slides when searching for a slide that covers a given keyword. This helps in getting the most relevant results first. With the increasing size of course materials gathered online, a user looking to understand a given concept can get overwhelmed. The standard way of learning and the concept of "one size fits all" is no longer the best way to learn for millennials. Personalized concept recommendation is presented according to the user's background knowledge.

Finally, the contributions of this dissertation have been integrated into the Ultimate Course Search (UCS), a tool for an effective search of course materials. UCS integrates presentation, lecture videos and textbook content into a single platform with topic based search capabilities and easy navigation of lecture materials.

ANNOTATION OF MULTIMEDIA LEARNING MATERIALS FOR SEMANTIC SEARCH

by
Sheetal Rajgure

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Department of Computer Science

December 2017

APPROVAL PAGE

ANNOTATION OF MULTIMEDIA LEARNING MATERIALS FOR SEMANTIC SEARCH

Sheetal Rajgure

Dr. Vincent Oria, Dissertation Advisor — Date
Professor, New Jersey Institute of Technology

Dr. James Geller, Committee Member — Date
Professor, New Jersey Institute of Technology

Dr. Dimitri Theodoratos, Committee Member — Date
Associate Professor, New Jersey Institute of Technology

Dr. Frank Shih, Committee Member — Date
Professor, New Jersey Institute of Technology

Dr. Pierre Gouton, Committee Member — Date
Professor, Universit de Bourgogne, Dijon, France

Dr. Roger Zimmerman, Committee Member — Date
Associate Professor, National University of Singapore

BIOGRAPHICAL SKETCH

Author: Sheetal Rajgure

Degree: Doctor of Philosophy

Date: December 2017

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2017
- Master of Science in Computer Science,
  New Jersey Institute of Technology, Newark, NJ, 2009
- Bachelor of Engineering in Instrumentation & Control
  University of Pune, India, 2002

Major: Computer Science

Presentations and Publications:

Sheetal Rajgure, Krithika Raghavan, Vincent Oria, Reza Curtmola, Edina Renfro-Michel, Pierre Gouton, "Indexing multimedia learning materials in ultimate course search.," *Content Based Multimedia Indexing*, 1-6, 2016.

Sheetal Rajgure, Vincent Oria, Krithika Raghavan, Hardik Dasadia, Sai Shashank Devannagari, Reza Curtmola, James Geller, Pierre Gouton, Edina Renfro-Michel, Soon Ae Chun, "UCS: Ultimate course search," *Content Based Multimedia Indexing*, 1-3, 2016.

Sheetal Rajgure, Vincent Oria, Pierre Gouton, "Slide localization in video sequence by using a rapid and suitable segmentation in marginal space.," *Color Imaging: Displaying, Processing, Hardcopy, and Applications*, 2014.

Duy-Dinh Le, Xiaomeng Wu, Shin'ichi Satoh, Sheetal Rajgure, Jan C. van Gemert, "National Institute of Informatics, Japan at TRECVID 2008," Text Retrieval Conference in Video 2008

To my beloved husband Neeraj, my son Nisheet and my entire family for always encouraging and supporting me.

ACKNOWLEDGMENT

I thank my Dissertation Advisor Dr. Vincent Oria, for his encouragement, support and guidance throughout my research. I am grateful for all the time he spent on providing ideas and comments to improve my work.

I would like to thank the Committee members, Dr. James Geller, Dr. Pierre Gouton, Dr. Frank Shih, Dr. Dimitri Theodoratos and Dr. Roger Zimmermann, for agreeing to serve on my Dissertation Committee and providing their valuable comments and advice. I would like to thank Dr. Pierre Gouton for the collaboration and guidance. I am grateful to Dr. James Geller for providing his help on the writing.

I would like to thank the Department of Computer Science at New Jersey Institute of Technology, for providing financial support during my PhD. I would like to thank Ms Angel Butler for providing exceptional help throughout this time.

I would like to thank the entire team of iSecure, Dr. Vincent Oria, Dr. James Geller, Dr. Soon Ae Chun, Dr. Reza Curtmola, Dr. Edina Renfro-Michel for their collaboration and valuable comments. I thank our UCS team, Krithika Raghavan, Hardik Dasadia, Shashank Devannagiri, Animesh Dwivedi and Hariprsad Ashwene, for their help in implementing the Ultimate Course Search (UCS) application. This work has been partially supported by NSF, under grant 1241976.

I would like to thank my lab mates, Ananya Dass, Souvik Sinha, Jichao Sun, Xiguo Ma, Arwa Wali, Cem Aksoy and Xiangqian Yu, for discussions and extending their help.

I would like to thank my family for their sacrifices, love and support for the entire time.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

Traditionally, courses were designed for face-to-face learning, where students and instructors meet on campus. The introduction of modern communication technologies, such as the Internet and email, offers an opportunity for the instantaneous exchange of thoughts and ideas. This has fostered the evolution of electronic learning, also called e-learning or online learning. A student may not be able to traditionally attend school for various reasons. E-learning relieves students from being physically present on-campus to receive the lecture at fixed time. The flexibility to take courses at their convenience allows them to complete their education. E-learning is a gateway for students across the globe to receive knowledge from online courses provided by experts.

In the last decade, e-learning has become a popular mode of instruction. Several universities offer online courses containing videos, slides, electronic textbooks and materials which are available centrally on a website, allowing students to access them anywhere at their convenience.

Massive Open Online Courses (MOOCs) like Coursera [35], MIT open courseware [64], Udacity [84] and edX [11] are storing large quantities of learning materials. Online lectures, to some extent, can emulate traditional classroom environments. Millions of students are registered for these courses and are taking advantage of these online courses to enhance their knowledge.

1.2 Problem Statement

Coursera has collaborations with more than 140 institutions; edX has more than 60 member institutions and provide hundreds of courses and specializations in different

areas. As there is a massive amount of content, navigating and effectively searching for a topic within the course material has become increasingly difficult. If a user is searching for one specific topic covered in a course, he or she might have to browse through the entire lecture material to find it.

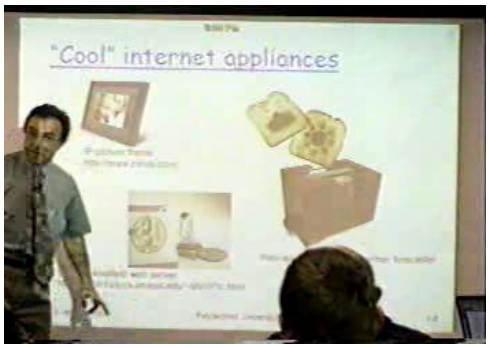
Online courses by an institution are powered by learning software usually hosted on a website. Each course contains sections, which contains links to the lecture videos, respective presentation slides and some reference materials. These courses are typically structured to navigate the course material sequentially. If a student wishes to study a specific concept in the course, the current keyword-based search tool fetches the entire video or slide which are not necessarily related to the required topic. This limitation overwhelms the student with the large number of search results and must go through entire section of the online material, i.e., presentation slide, lecture video or the textbook to look for it. The current MOOCs lack the feature of topic-based priority search. An interface capable of searching for a topic and providing relevant results in the lecture material that are meaningful to the search would be ideal.

A student preparing for a course views videos and other materials on the e-learning site, refers to a textbook recommended by the instructor and searches and reads resources present on the Internet. The absence of an integral portal containing all the learning materials and other resources, including the textbook in electronic format, would save time and help the student to grasp the information quickly. Users could search for a topic and the resultant material would be specific slides, segments of the videos and most relevant pages of the electronic textbook associated to the topic.

To create a topic-based search on lecture materials, the content needs to be indexed. Given the amount of learning material present, annotating the content manually would be a resource-intensive and tedious task. Therefore, automating the annotation process will be beneficial. Segments of a video could be annotated to

represent the information of each presentation slide covered in that video segment. The lecture videos are shot under different conditions: some of them are recorded in the classroom while others are a screen capture using screen recording software.

(a) Case I - Full screen Image



(b) Case II - Occlusions and poor quality

(c) Case III - Slide covering small part of video frame

Figure 1.1 Types of lecture videos.

Many videos have inadequate illumination, the presence of instructor and audience, partial slides and poor quality. All these challenges make it difficult to detect transition changes accurately (Figure 1.1). Case I shows a video, having poor quality and occlusions, such conditions often generates false positives due to obstructions. Case II shows a slide, that covers only 40% of the entire frame. Since

the slide covers a minor part of the frame, any change in background or speaker movement can contribute to false detection.

Optical Character Recognition (OCR), recognizes the characters in the image. Using OCR on the video stream directly can yield poor results and the processing time is slow. The detection results are entirely dependent on the OCR software and the false rates are high. In cases where slides contain only images or figures, OCR does not work.

Automatic speech recognition (ASR) is still an active research area. The problem with ASR is that the accuracy of generating keywords is entirely dependent on the speech processing engine and the accuracy is generally poor. ASR systems are often trained on native English accents. The false detection rate is relatively high for non-native speakers of English. Such systems have to be trained for different speakers and manual effort is needed to correct the speech in these cases. For foreign languages, a new model needs to be added to the ASR systems with the vocabulary and needs to be trained. Many commercial products like dragon also require the user to be trained for different speakers.

1.3 Annotation in UCS

Annotating learning materials has various challenges. Different media are used for teaching, such as slides, videos and textbooks. However, the annotation process is different for each. The learning materials were processed separately to annotate the content (Figure 1.2).

1.3.1 Slide

The following steps are performed to annotate the presentation slides:

Keyword extraction: Keywords are extracted from the slide text.

Structure extraction: Region information of keywords is extracted, based on their

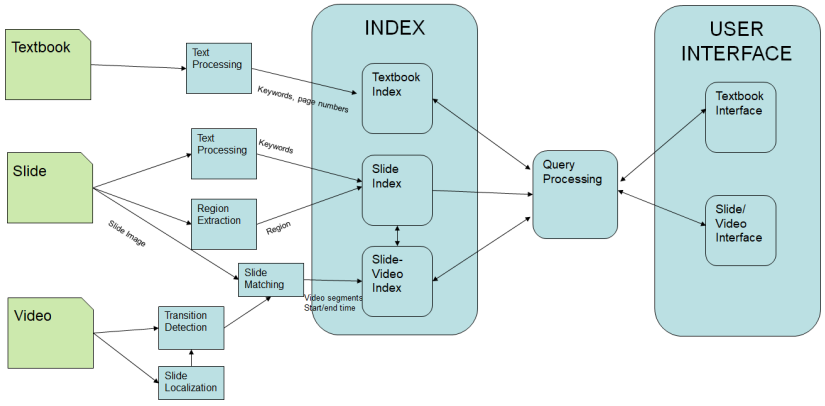


Figure 1.2 Annotation in UCS.

appearance in the region, which helps to present results based on priority.

The slide annotation contains an index of the keywords and locations of the slides, where the keywords appear.

1.3.2 Video

Video is essentially a series of frames. To annotate the video automatically, individual frames are analyzed. The following steps are performed at the pre-processing stage.

Slide localization: The slide is detected and extracted from the video frame for the classroom videos.

Shot transition detection: Features extracted from the slide frame help to determine transitions in the lecture video.

Slide and video mapping: Once the transitions are detected, the video segments are mapped to the respective slides that appear in the video segment. The videos are annotated using the mapping between slides and video segments (slide-video index).

1.3.3 Textbook

The textbook is annotated by extracting the keywords and associated page numbers. The textbook index contains information related to page numbers and keywords.

1.4 Outline of Dissertation

The history and evolution of MOOCs, is discussed in Chapter 2. The related work done in content annotation over the years is presented.

Chapter 3 describes the processing step performed on screen capture videos. Screen capture videos often come along with metadata which is not reliable many times. To create a mapping between the video and slide, the video is partitioned into segments such that each segment of video corresponds to one slide also called as transition detection. Background for shot boundary detection is presented, that includes the three steps, feature selection, feature similarity measure and shot boundary detection. Related work done in this area is also presented. The performance of state-of-the-art feature descriptors, namely Histogram of Oriented gradients(HOG), Wavelets, Edge Change Ratio (ECR), Scale Invariant of Feature Transform (SIFT) and Fast Fourier Transform (FFT), is compared. The goal of this study is to determine suitable descriptor for lecture videos.

Classroom recorded videos need special processing as discussed in Chapter 4. Classroom video recordings also record the surrounding classroom (e.g., the audience). It is necessary to extract the slide region (localization) to avoid the false transitions. The region extraction process consists of two steps. The first step towards localization is to segment the frame into clusters of region. This is done using k-means clustering, which involves clustering in R, G and B space. It is observed that many images like lecture video frames are not color predominant. It is advantageous to convert them to the different color spaces, namely DCT, Marginal, and grayscale, to determine the most suitable technique for segmentation. K-means clustering based on Tsai's moment

preservation technique for multiple thresholds is proposed. These thresholds are set as initial seeds. A comparative analysis of k-means clustering on DCT, marginal and grayscale space is presented. The slide is then localized (extracted) using some heuristics.

The annotation process of learning material in UCS is presented in Chapter 5. The indexes are set up differently for each of the learning media (slide, video and textbook). To prepare annotation of the videos, we use slide index as a base. The slide index is prepared by extracting the keywords from the slides. For textbooks, we use the rear index of the textbook to prepare the indexes. Videos are indexed with the information such as the timestamp corresponding to the slide.

Often the top results returned for a search do not explain the topic very well. A region-prioritized ranking mechanism is proposed, to prioritize the most relevant results according to the region of appearance. If the keyword appears in the title, more weight is attached to it than if it appears in the body. This makes sure that relevant results appear at the top.

The contributions of this dissertation have been integrated into Ultimate Course Search (UCS), a tool for an effective search of course materials. UCS integrates presentations, lecture videos and textbook content into a single platform with topic-based search capabilities on the actual content and easy navigation of lecture materials. Users can query the UCS system with keywords, and the system returns the links to the learning materials (slides, videos and textbooks) that cover the topic. Upon selecting the video option, the user can directly view the segment of video related to the topic. Similarly, the portions of slides and textbooks are displayed that cover the topic. Users don't have to navigate through entire lecture material by using UCS.

Chapter 6 covers personalization of learning concepts based on the knowledge of students due to the courses they have completed. In the case of e-learning systems,

personalization can mean adapting the content to the user learning preference or the knowledge level. The goal of personalized learning is to recommend the prerequisite concepts, to prepare the user for the given concept. Chapter 7 concludes the dissertation and proposes future work.

CHAPTER 2

E-LEARNING BACKGROUND AND RELATED WORK

The term "distance learning" has existed for quite some time. After the lecture materials became available online, the term "e-learning" was coined in 1999 as a short version of electronic learning. The purpose of e-learning has been always to make knowledge and resources available to the students who cannot manage it due to the time. E-learning is an equivalent offering of the classroom lecture for those who cannot physically attend the course. Below is a summary of how e-learning evolved.

2.1 E-learning Background

Distance education learning is categorized in two ways, synchronous and asynchronous learning. In synchronous learning, the course is scheduled at a specific time and all participants are present at the same time and in some way, it replicates the traditional classroom experience. Web conferencing, video conferencing, television broadcasting, internet radio, live streaming, telephone conversations and VoIP are all forms of synchronous learning.

In asynchronous learning, participants can access the material according to their schedule and convenience. Mail is one of the oldest correspondence mechanisms is based on asynchronous learning. Similarly, emails, chats, message boards and all the modern MOOC systems are based on asynchronous forms of learning.

2.1.1 Pre-history of E-learning

Previously, lectures were strictly in the classroom. Sometime in 19^{th} century, postal services became faster, and distance learning spread across Europe and the United States. The first such course was taught by Pitman in the1840s. He started a correspondence course on shorthand where he sent his assignment to his students by

mail and his students would send the completed assignment back. Throughout the 19^{th} century, distance learning was widespread. For example, students in Australia could take correspondence courses offered from London School of Economics. In correspondence courses, the materials were sent by mail and students could study without having to be present physically.[65].

In 1920, Pressey developed a testing machine. Users were presented with a question with 4-5 possible answers, the question number appeared on the device, and the user was given multiple choices as possible answers (numbered 1-5). When users pressed a key as a response to the question, their response was recorded by the device on the test sheet. After they finished, their final score was also noted on the test sheet. This was the first automatic grading. In the early 20^{th} century with the advent of technology, things changed as radio and television could be used as media of instruction.

In the 1950s, as television became popular as a medium of education, and there were not many teachers available to teach in United States, B.F. Skinner developed a learning system around this time which differed from Pressey's in some ways: he presented the material in chunks. The teaching machine was mainly a program, which was a combination of teaching material along with test items along the way of the material. The program was composed by either fill in the blank or workbook or in computer. The correct answer was revealed later. If the user selected the correct answer, that was reinforced; otherwise, the user studied the answer to learn the correct one for next time. This technique of teaching was called programmed instructions.

In the late 1950s "Computer Aided Instruction" or "Computer Assisted Instruction" (CAI) was introduced in elementary schools. It was a joint effort between educators of Stanford University and IBM. In CAI systems, information was combined with drill and practice sessions. During this time, obtaining and maintaining computers was difficult due to which these systems were limited. "Programmed Logic

for Automated Teaching Operations" (PLATO) [70], was another early CAI system started at the University of Illinois and used for higher education. It included drills and practice exercises, and it consisted of a mainframe computer that had around 1000 terminals to support the students. By 1985 there were 100 PLATO systems that were operating in United States. Around 40 million users were taking advantage of PLATO systems from 1978-1985. These systems paved the way for communication between users, similar to email technology developed later.

In 1969, ARPANET was born as a research project at DARPA, and UCLA was connected as a host to ARPANET. Computers were added to the ARPANET during the following years and gave birth to the Internet. In 1972 the initial application of electronic mail was developed. The simple application had read, file, respond and forward the message. This made the communication between people much faster and easier.

In 1969, Open University [7] in the United Kingdom was another initiative that took correspondence learning to the next level by using multimedia. Open University the first successful distance learning university. When email became available, Open University began to communicate the course material through it. This was a breakthrough in the field of education. The communication was instant and would no longer take days with postal mail. Some people took advantage of this technology and accessed the learning materials for free. Through the 1970s and 1980s, enrollments increased steadily and more courses were introduced.

In the 1970s, with the invention of GUI and mice for computers, the new era of e-learning began with Computer Based Training (CBT). The training to any individual was given through a computer. The first CBT was developed in 1976 [87] at New Jersey Institute of Technology (NJIT). Electronic Information Exchange system (EIES) was developed. The students could have stored class discussion in an asynchronous manner. They concluded that discussion in such a manner proved to

be informative for both instructor and students, with students actively participating in the discussion.

When the first personal computers were introduced by Altair, Apple II and IBM PC, individuals gained access to the computers for their personal use. CBT gained popularity around this time. Users needed to install the program stored on media, such as floppy disk/ CD-ROM/ DVD. With the CD-ROM and DVD, more information could be stored in the form of instructional material.

With the Internet available to people, online courses gradually picked up pace. The first fully online university, the University of Catalonia, was founded in 1994 in Spain [10]. From 1990 to 2000, many of the Learning Management Systems (LMS) that remain popular today were developed. Institutions used an LMS to deliver content online. The students could submit their assignments and instructors could keep track of the grades using LMS. In 1997, CourseInfo LLC launched "Interactive Learning Management" at Cornell University, Yale Medical School and University of Pittsburgh. Later, CourseInfo LLC merged with Blackboard Inc [1] to launch Blackboard Learning System.

In 2001, MIT initiated "MIT Open Courseware" (OCW) [64]. OCW opened the set of 32 courses in 2002 to the public. Various course materials, such as the syllabus, an introduction about the course and video lectures were hosted online. Additional materials, such as power point presentations, pdf files and others were also uploaded where applicable. Each course was divided into several lectures, having their respective supplementary materials.

In 2002, Moodle [6] was introduced and remains one of the most popular LMSs. In the 2000s, businesses started to present e-learning courses to train their employees with new skills to improve their knowledge and in turn help companies on their projects.

From 2000 to 2008, online education was widespread in most of the countries throughout the world. Social media, iTunes U and MOOCs took the online education to a totally new level by giving students a platform not only to learn at their convenience but also to communicate with the instructor and other students instantaneously and most effectively.

In 2004, Salman Khan, a Harvard and MIT graduate, started remotely tutoring one of his cousins interactively using Yahoo Doodle images. That was an instant hit among his relatives. To make better use of time and allow flexibility, Khan posted the videos on YouTube. After becoming popular with many students, Khan established Khan Academy and began working full-time on that. A typical tutorial has a video that shows step-by-step doodles and diagrams on an electronic blackboard. Other than tutorials, the website also has features such as progress tracking, practice exercises and a variety of tools for teachers in public schools There are more than 4500 tutorial that cover different academic fields. The organization is supported through donations and has more than one million subscribers [4].

In 2007, Apple announced the launch of "iTunes U" via its digital content store iTunes [5]. This service was aimed at students of various universities and they were given access to their university's video and audio content. Each member university creates its own iTunes U site, which facilitates searching for material. Many colleges and universities in various countries, such as the United States, the United Kingdom, Australia, Canada, Ireland and New Zealand, offer iTunes U that includes lectures, language lessons, lab demonstrations and more. Anyone with an Apple device or using iTunes software was able to access the content with ease. As of 2011, Open University in the UK set the record for the most downloads having reached 40 million downloads. In early 2013, iTunes U crossed the mark of one billion downloads from more than 800 institutions. From 2008 the new era of MOOCs had begun, and it took e-learning to new heights. The next section talks more about MOOCs.

2.2 Massive Open Online Courses (MOOC)

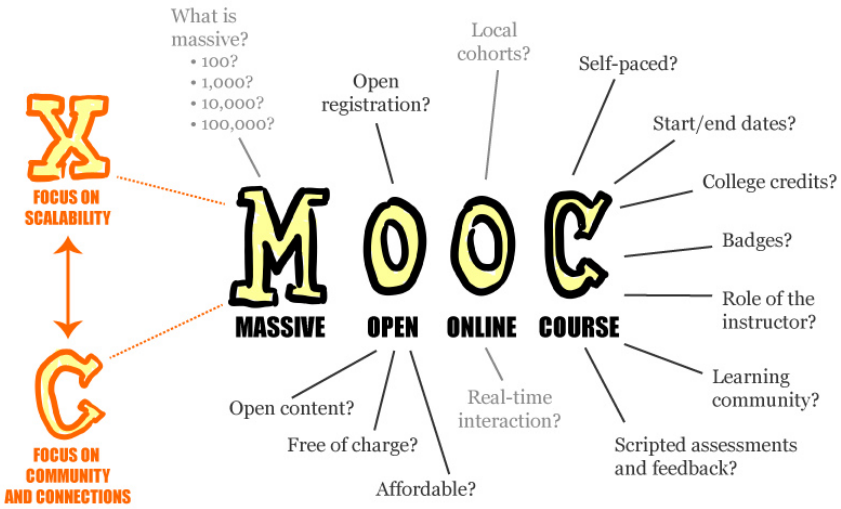


Figure 2.1 Massive open online courses (MOOC), types and criteria.

A MOOC is an online course aimed at massive participation from students across the globe via web [66]. The courses offered have course material similar to classroom lectures and also the video of the instructor explaining the content. Students are provided with an assessment comparable to the classroom lectures. An essential aspect of MOOC is the network, people across the world can discuss the course contents with each other. There is no prerequisite to join the course, such as completion of any specific degree. The aim is to give free (sometimes paid) education to students who are often unable to attend school. In MOOC systems, learning and success are entirely dependent on one's participation.

2.2.1 MOOC is Massive

Massive Open Online Course as the name suggests is an online course that is massive in the sense of participation of students. Traditional classroom courses are limited concerning the number of students that they can accommodate. On the other hand, the massiveness in MOOC systems is regarding the number of students taking advantage of this course. Users across the globe can access any material without any

restriction. Some MOOC have few hundreds of students enrolled while others have more than thousands of students enrolled for every course.

2.2.2 MOOC is Open

Originally, the creators had a different idea of openness of MOOC. The creators felt that the MOOC should be open in the sense that anyone can access, use it and edit the material. The materials should not be copyrighted or property of specific institution or person. The content should be modifiable so that more people share their experiences and thought to make it better.

Today that concept has changed, such that most of the courses in Coursera or edX are copyrighted, and users are not allowed to modify those. The materials are available only for the duration of the course, this is not what was initially intended, and over the years the purpose has changed regarding the openness. In short, today MOOCs are open with some restrictions.

At the present time, open means several things. The MOOCs are open in the sense that anyone can access the material and the course is free. However, some of the MOOCs, such as Coursera or edX have specialized courses or certification courses that are offered for some fee. The meaning of open is a course that is available at no cost without any prerequisite. In other words, anyone with a degree, either in high school or college can take any course, and there is no minimum qualification set for any course.

2.2.3 MOOC is Online

MOOC is offered on the web so that students can take advantage of the course remotely from any part of the world. MOOC also enables the students to share their thoughts and ideas with each other about the course and allows an instructor to reach out to everyone, similar to the traditional class, but at a substantially large scale.

2.2.4 MOOC is Course

In the field of education, a course means a unit of teaching a specific area with predefined objective or goal, which is often taught by one or more instructors. The course usually has some definite duration, with start and end dates marking the course duration. Upon completion of the course, the instructor determines the progress of each student by giving them a grade which marks the end of the course.

MOOCs are a courses in some sense. Each course has a fixed start and end time, typically lasting 4-6 weeks. Every week there is some specific chapter or topic covered and students can control their pace. In other words, users can choose when to view lecture material according to their comfort.

In MOOC the completion of the course is entirely up to the student. MOOCs provide regular assignments, quizzes or projects to students to help them understand the topics better and set some milestones along the course completion. Every assignment has some deadline associated, similar to the traditional classroom course. It is difficult for an instructor to evaluate the work considering the massiveness of students. Hence, the assignments are assessed by software or peers.

2.2.5 Types of MOOCs

MOOCs are broadly divided into two categories namely cMOOC and xMOOC [2], [3]. cMOOCs work on connectivism philosophy of MOOC that was proposed in earlier versions of MOOC. In cMOOC, the learners create their own goal, where the learners are not evaluated, but they control their learning. The starting point of the connectivism is an individual learner, the learner has several connections, and the information is exchanged through the network. The participants act both as teachers and students. The learning is through collaborating with each other through a social network; the learners are connected for discussions and to work on a joint project and thus create a platform for future learning.

There exist some MOOC types worth mentioning, such as smOOCs and bMOOCs. sMOOC is the small open online course that includes a small number of students. bMOOCs are the blended or hybrid MOOCs where the course is taught in the class as well as available online; this gives students flexibility regarding time.

2.2.5.1 cMOOC. Downes identified four key design principles for cMOOCs:

Autonomy of the learner: Here the learners choose what skill they want to learn. There is no set format or outline of the course.

Diversity: Range of people who participate and their knowledge

Interactivity: Co-operative learning, communication between participants leading to knowledge exchange.

Openness: Openness is regarding free access, free content, and assessment

Thus in cMOOCs, the learning occurs due to exchange of information between the participant, unlike xMOOC where the knowledge learning results from an instructor who is an expert in that field.

2.2.5.1.1 *Design Features for cMOOCs*

Today, cMOOCs take advantage of some of the following techniques:

Social Media: For cMOOCs, the first important requirement is the network. The network is where one can start sharing the information. These may include web conferencing tools, such as Google Hangouts or Adobe Connect, streamed video or audio files, blogs, wikis, learning management systems, such as Moodle or Canvas, Twitter, LinkedIn or Facebook, all enabling participants to share their contributions.

User-driven curriculum: There may be moderator assigned to start a particular topic, but the content is driven by the users who participate in the discussion.

Assessment: There is no instructor or authority available for assessment. Even

though some of the users get an advantage of sharing knowledge from a more experienced user, it is up to the user to decide the completion of course.

2.2.5.2 xMOOCs. The xMOOCs, on the other hand, are based on the idea of traditional classroom courses. In xMOOC, there are one or more instructors, supported by the TAs and several students are enrolled for the course. The lectures are prerecorded videos, accompanied by other learning material. This form of MOOC has a more controlled layout of the course, and the topics covered in xMOOC are predefined. The students are assessed with quizzes, assignments and projects. There are peer assessments and discussion forums where students can participate and share their thoughts.

2.2.5.2.1 *Design Features of xMOOC*

Software: xMOOCs use specialized software that acts as a platform for the registration of vast numbers of participants, storing and streaming of content and assessment

Short Video lectures: xMOOCs use the recorded videos that are hosted online. Usually, each course spans for 8-13 weeks. The research shows that the human attention span is limited; due to this people lose interest if the video lectures are too long. In xMOOCs, the video lecture is broken down into smaller segments of 15-20 minutes to help students retain their interest. These videos are shot either using a camera covering the class or by the instructors themselves using web-camera or screen capture software.

Computer Assessment: Students complete an online test or quiz and receive immediate computerized feedback. The course grade is earned after completing the online test or project. Most xMOOC assignments are based on multiple-choice questions, but in some cases, MOOCs also used text fields for participants to enter

answers, such as coding in a computer science course, or mathematical formulae, and in one or two cases, short text answers, but in all cases, these are graded automatically.

Peer Assessment: Some xMOOCs divide students into a small group where they assess the assignments randomly submitted by their peers. Sometimes it may cause a problem due to a different level of expertise these users may have.

Supplementary Materials: Often there is some supplementary material along with the video lectures to help students learn better, such as Slides, audio files, URL link to some of the reference material, a discussion forum where the enrolled students can communicate and discuss problems faced.

Certificates: After successful completion of the program, most xMOOCs award the students with some recognition in the form of certificate or a score learning analytics.

Learning Analytics: xMOOC platforms can collect and analyze the data about how students learn which may be necessary to determine the success rate of the MOOCs.

2.2.6 Emergence of MOOC and Features

The first MOOC was launched in 2008, named "Connectivism and Connective knowledge" CCK8 [40]. The course was free and available to anyone. This course was open, and anyone could join, modify the content. Students were both teachers and students. The primary objective was to encourage collaboration between different people. Students share their knowledge through networking with other students. Here there is no predefined outline of the course. The students ask questions, and the knowledge is transferred within the network. The students were engaged in learning by using different platforms, such as Facebook, wiki pages, blogs and forums. Around 2,200 people joined, and many of them created blogs. The CCK8 comes under the cMOOC category.

Peer to Peer University (P2PU) [8] was established in 2009. P2PU is aimed at people sharing their knowledge with each other. Anyone can either create or take a course. The idea is that students learn better socially, and the results are often better. The participants of the class communicate with each other synchronously on tools, such as Skype or IRC and they can interact asynchronously on the P2PU website.

Three professors at Stanford, Andrew Ng, Daphne Koller and Sebastian Thrun paved the way for the modern MOOC that exists today. Ng started in 2007 with a primitive approach; he recorded his lectures and put them online. His idea was to reach out to students who did not get a chance to have education at Stanford. To his surprise thousands of students took advantage of his course on machine learning.

Koller felt that passively listening to the entire video might not be that interesting and that students could lose out interest quick. She felt that if the videos were divided into chunks of shorter segments, it might interest the students further. Ng and Koller collaborated with each other to improve the quality of learning. They felt that students often learn better when they discuss the material, that way they know several possible solutions. Taking the idea from social networking, the two professors decided to integrate the discussion forums to the software built by them.

IIn 2011, Stanford University launched three courses, first was offered by Sebastian Thrun and Peter Norvig from Google, launched "Introduction to AI", this course was a huge success. As soon as it was announced, the enrollment rose to 160,000 students to everyone's surprise. Shortly after the success, two other professors Ng and Widom launched two more courses which were also a huge success.

Based on the popularity of the courses, Thrun established Udacity [84], Ng and Koller launched Coursera [35]. Coursera announced a partnership with University of Pennsylvania, Princeton University, Stanford University and the University of Michigan. In 2013, Udacity launched its first for-credit course in collaboration with San Jose state university.

Anant Agrawal, one of the MIT professors, was inspired by the success at the Stanford University. With the commercialization of online education, he created a not-for-profit initiative MITx. Harvard joined the group and MITx was renamed to edX [11]. Several other universities joined too. In 2013 Google announced a partnership with edX. Google and edX will work on the research dealing with how students learn. edX launched a joint open source platform that is under GPL license. Originally the MOOC was designed to be open in the sense that the material is freely available and without any copyright issues, but with Coursera and edX the materials were copyrighted.

2.2.7 Features of Current MOOCs

Features of the popular MOOCs namely coursera [35], Udacity [84] , edX [11] and Udemy[9] are as follows:

2.2.7.1 Coursera. Coursera was established by Ng and Koller in 2012. Following are some of the technical aspects:

Platform Used: Coursera runs on Nginx web server on Linux OS on Amazon web services with the primary stack in Scala on the Play Framework. Data is stored on Amazon S3.

Specialization: Coursera offers Specialization with fee, but one can audit the course for free.

Assignments: The courses are typically 10-12 weeks. There are quizzes, weekly exercise, assignments and project.

Course Outline and Features: The course is split into weeks, one chapter every week, and each chapter is broken into small segments of 7-10 minutes. The videos are recorded using screen capture software. There are quizzes and assignments at the end of each week. There is a discussion forum where the students can discuss course-related problems. Students can also form a study group. Few moderators can

help to solve course difficulties.

2.2.7.2 Udacity. Udacity uses Wacom Cintiq tablets to make their presentations. The course is divided into several lessons, such as Coursera. The grading is automated. Students can take certification courses called Nano-Degree programs which are categorized according to their difficulty level. The lecture videos could also be downloaded. The lectures are uploaded on YouTube makes use of YouTube close captioning. The transcript is available on YouTube, and if the user clicks on the transcript, they can be directed to a segment of the video. YouTube uses the automatic speech recognition API (ASR API) from Google that creates the transcript automatically.

2.2.7.3 edX. edX differs from Udacity and Coursera in the way that edX is a not for profit organization and also it runs on the open source platform. In edx, the course is divided into weeks of study, such as Coursera and Udacity. The videos are of shorter duration. On the front-end, there is a provision to see the transcript along with the video. As the video progresses, the text in the transcript is highlighted and the user can navigate the video through the speech transcript which plays the video from that point. However, instructors are required to upload the speech transcripts.

2.2.7.4 Udemy. Udemy offers paid or free course depending on the instructor. Using their platform instructors can upload their content in the form of videos, PowerPoint, pdf, audio and zip files. Udemy provides instructors to create a course advertise it and earn profit from the tuition of students enrolled. Courses are split into various sections. Lectures are usually video, but they can also include audio, text, and presentation slides. Users can access the related material, discussion forums.

Every course has a web page with a brief description of the course, user reviews, and sometimes a video introduction.

2.2.8 Completion Rates

Completion rates are anywhere between 7% - 10% which is due to some of the factors such as, the course is too easy or difficult, lecture fatigue, lack of organization of the course, many students want to take an overview of the course rather than completing it.

2.3 Related Work: Annotation of Learning Material

Annotating the content is to tag the content in such a way that the content can be searched using the tag. The annotation of any material defines the content. The content usually is in the form of text (slides), visual (video) or audio (speech of instructor).

2.3.1 Manual Annotation of Lecture Videos

Earlier when the technology wasn't advanced, the annotation of the content was done manually. Manually annotation is a tedious work and requires time and labor. This section presents some of the related work done in this area.

One way to annotate the lecture materials, is to attach some meta-data to the lecture video as described by Jesse et al. [51]. But the metadata usually provides a summary of the entire video and most of the time this is done manually. Annotation on the metadata limits the search capability to only the keywords appearing in the metadata than on the entire lecture video.

Classroom 2000 project [12] introduced a presentation tool called ClassPad that uses an electronic whiteboard to present slides and allow annotation by the instructor. The instructor annotates the GIF image of the slide with a pen. Students are also provided with the similar interface where they can take their notes. Both the

approaches rely on some form of hardware or software systems that have to be used to prepare the recording. The problem with the approach is that manually editing the timestamps, is a tedious task and takes a significant amount of time.

In Microsoft research Annotation System (MRAS) [21] the authors describe an annotation tool that lets users annotate the content played. In this system, the student can login to watch the class lecture; they are provided with an interface where a user can enter questions or comments related to a specific part of lecture video. These comments are saved so that when other user logs into the system, they can watch the lectures along with the comments that were entered into the system prior by other students who viewed the lecture before them. The questions are linked to the content, and as the user progresses watching the material, the comments are highlighted in that section of the course material.

2.3.2 Automatic Annotation of Lecture Videos

For lecture video, automatic annotation means extracting the semantic information either directly from the text that appears on the video frames using some character recognition method or to map the slides with video segments. Another approach is to convert the audio stream of the instructor into text and use the text to annotate the video.

2.3.2.1 Annotation Using Text Extraction Techniques. One way to extract the text from the video stream directly is by using Optical Character Recognition (OCR). OCR is an electronic conversion of images of typed, handwritten or printed text into machine understandable text. In OCR, first the computer is trained with samples of characters on test data, and then it recognizes those characters. Early research has been done in this area.

Liška et al. [57] use OCR tool on the extracted frames for text recognition. OCR is performed on every extracted slide frame which amounts to a lot of computation

time. They do not use text localization process to identify text regions. Therefore, the recognition accuracy of their approach is much lower.

TalkMiner [15] offers search capabilities similar to UCS but on talks. Talk-Miner provides an interface where the user can browse through the talks and search for specific talks using keywords. The keywords are extracted from the video directly using optical character recognition (OCR). Thus, the accuracy of keywords is entirely dependent on the accuracy of OCR. The text extracted is used to build indexes to support the search. OCR techniques are slow and not always reliable especially when the quality of the video is low; this can impact the search capabilities. TalkMiner is proposed on general talks, and their numbers are limited compared to course lectures given at higher education institutions.

Yang et al. [92], presented an approach for automated lecture video indexing based on video OCR technology. Instead of giving the whole frame as an input to OCR and then detecting the text, the text is localized from the frame, such as slide title and subtitles and those sections are given as an input to OCR.

Second way to annotate the videos is by aligning the video segments to the slides.

In the work [77], authors mapped the videos with the slides to index the content. They manually edited the time stamps of the transition points for synchronization.

Another approach is that used in BIBS lecture webcasting [78] system that provides the user with a tool to review the lectures. The lecture is recorded using their software. To map the slide and the video, they use a plug-in to record the slide time codes automatically. Otherwise, they enter it manually. If the lectures are recorded with software, then this approach works fine, but if we want to include some old lectures or any other videos not recorded with the software, then the approach fails.

Mukhopadhyay et al. [67], propose an authoring system. First, the transitions are found by binarizing two frames after a chosen interval. The difference between two frames is computed regarding pixels, which is compared against a fixed threshold. For the matching step, both the slide and video frame are clipped and binarized. After that, both the images are dilated, and again the number of black pixels is computed as a measure of similarity.

Hunter et al. [50] proposed Synchronized Multimedia Integration Language(SMIL), a technique to index a multimedia presentation archive. In this method, the authors prepare a system where the instructors upload the slides in the pdf format. The recorded video is mapped to the uploaded slides. The video frames are binarized, and pixel difference technique is used to find the transition point. The major problem in above two approaches is that it relies on pixel-based information which is often noisy and unreliable.

Mapping video with the slides is a two-step process. The first step is segmentation of video and localizing of a slide in a video and secondly to match the video with the slide. To map the slides and video, we need to segment the video such that each segment contains the talk about each slide.

Image processing techniques are used to identify the transitions automatically. In a lecture video, each video segment corresponds to one slide of the power-point presentation. The method to detect the transition is also termed as shot boundary detection. Extensive research has been done on shot boundary detection.

2.3.2.2 Annotation Using Speech Transcript. Another approach is to use the audio layer as suggested by Kamabathula et al. [52] and Repp et al. [76]. In the work proposed by Kamabathula et al. [52], a video browsing tool is developed that uses speech transcripts to generate keywords for indexing. The audio track usually produces a considerable amount of keywords, and many of them do not appear in

the slides. For the cases where we do not have access to the speakers to train the software, it becomes challenging to get correct results. In the approach suggested by Kamabathula et al. [52] and Repp et al. [76], slide transitions were detected, by recognizing the characters using OCR technique.

Yang et al. [91] used Automatic Speech Recognition (ASR). These systems are based on machine learning and by default trained in American English and needs an intensive training for non-native English speakers. In the lecture browser, the authors train their system for various speeches of non-native speakers. This process requires extensive manual training.

In systems, such as edX, the users are required to upload their transcript; this leads to an additional step to align the transcript either by themselves or using paid software.

CHAPTER 3

ANNOTATING SCREEN CAPTURE VIDEOS

Videos are mainly a series of consecutive images called frames. A "video shot" is defined as the sequence of continuous frames shot without any interruption by a single camera. These shots are grouped into scenes based on location. Shot boundary detection techniques are used to segment a video temporally into smaller segments based on camera movements. In an educational video, each shot refers to a slide of the PowerPoint presentation. A video consists of several frames, and selected frames called key-frames are picked to represent video shots.

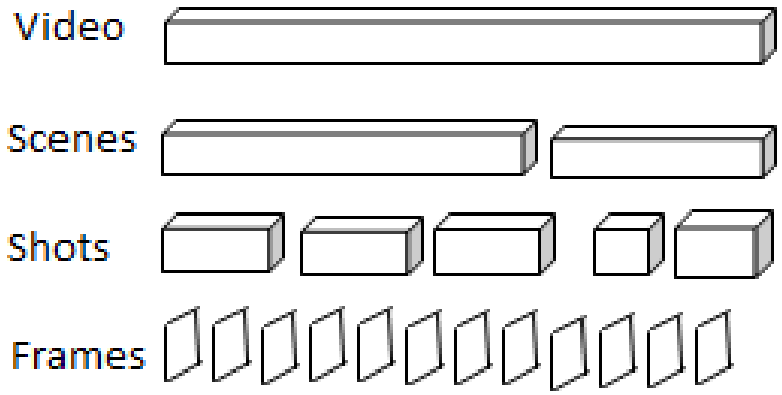


Figure 3.1 Decomposition of video into scenes, shots and frames.

Educational videos, capture not only the content of PowerPoint slides but also the instructor's explanation of the slides. The videos need to be segmented based on the slides obtained in a video recording. Knowing the slide transitions in the video is the first step towards annotating the lecture video. We must find the time in the video recording when the slide first appears and the duration for which the slide is covered in the recording. Each part of the video is referred to as a video segment

associated with a slide. Video index generates a mapping between the segments of the videos and slides in the corresponding lecture presentation.

It is essential to select best features to identify shot boundaries, in an educational video. Choosing an ideal feature to represent the video frame, remains a problem in instructional videos, due to the nature of the lecture video. Learning video recordings capture presentation slides whose template remains unchanged, except the text or figure. Therefore, detecting a transition is challenging.

Two kinds of videos are commonly used: (a) Videos shot in a classroom ((Figure 1.1) Case II and Case III) and (b) Screen capture videos (Figure 1.1 Case I), where the instructor uses dedicated software to capture the computer screen. Unlike the screen capture video which may be accompanied by metadata, generated by the recording software, the classroom video is captured by regular cameras and are more difficult to process. This chapter describes the processing and annotation in the case of screen capture videos. Chapter 4, presents the details of annotation in the case of classroom videos.

Screen capture videos are usually of good quality. In some cases, it was noticed that the metadata was not consistent, and some transitions were missed. Therefore, instead of relying on metadata, image processing techniques were used to identify the transitions. The following steps are carried out to align the videos with the slide.

First, the background details on shot boundary detection and the related work on general videos are presented. Next, the commonly used descriptors on educational videos are described. These descriptors can be used to segment the videos and provide the time when the transition occurs. An additional step of slide matching is proposed for aligning slides to videos; in case the slides were skipped. Slide matching also serves as a verification step for the transitions detected. Finally, the slide-video index is proposed that serves as the mapping between slide and videos.

3.1 Shot Boundary Detection Background

To associate a slide with a video segment, we need to find the transitions in the video. The technique to detect transitions in a video is known as shot boundary detection. In an educational video, each transition corresponds to the slide change. As opposed to shot transitions in generic videos, the transitions in a lecture video are based on slide change rather than camera motion. Therefore, the descriptors need to be invariant to the camera motion and animation effects in the slide presentations, such as text animations and capture different types of transition effects. The shot transitions for lecture videos can be of various kinds. The usual slide transition effects are *cut*, when there is an abrupt change of slide, *gradual*, when a slide fades in while the previous fades out, *dissolve*, where the previous slide disappears within the next slide, *wipe*, where the next slide gradually wipes out the previous one. The slide transition length can be from a few frames to hundreds of frames; these challenges make it difficult to select suitable descriptors for lecture video.

The shot boundary detection process is divided into three parts. The first step is to create a feature vector for each video frame. The second step is to define a similarity function that can capture slide transitions, by comparing feature vectors chosen in the feature selection process. The final step is the shot boundary decision to declare a shot boundary, if the distance between two video frames is above some set threshold.

3.1.1 Feature Descriptor Selection

The feature descriptor selection is the first shot boundary detection. Kikukawa et al. [53], proposed feature descriptor based on the change in intensity in the images. But later, several techniques were developed in this area and can be broadly categorized into the following:

3.1.1.1 Pixel-Based Methods. Early methods in the field of shot transition detection were pixel-based, where the change in the number of pixels between two consecutive frames was computed. If the number of pixels that do not match is above some threshold, a shot boundary is declared. One of the pixel-based approaches proposed by Nagasaka and Tanaka [68], used techniques like sum of difference of intensities of pixels. Otsuji [71], used pixel-based inter-frame distance. This method is susceptible to any motion. In the work proposed by Choubey et al. [31], the authors also focus on pixel-based shot boundary detection.

Zhang et al. [95], made some improvements by making use of 3×3 averaging filters to reduce the motion effect in the video. The image is filtered, before comparing the pixels from two different frames. A block matching scheme was presented by Shahraray et al. [81], to compensate for the motion. The image was divided into 12 blocks, and the pixel intensities between two frames were compared. The block matching scheme provides more room for any movement instead of comparing pixel at a specific location in one frame to the pixel in the same position in another frame. In the work proposed by Ngo et al. [69], pixels from a specific part of the video frame were subsampled. Pixel-based methods are fast regarding computation, but they are susceptible to movement of camera or change of illumination and can produce many false positives.

3.1.1.2 Histogram-Based Methods. A better approach than pixel-based approach is that of computing histogram of one frame and comparing it against another. The histogram approach considers the distribution of pixels rather than the location of pixels. The histograms could be calculated either on the intensity (grayscale) of the pixels or their color information [47]. Histograms are very efficient concerning computing time and are insensitive to small camera movements. The histogram does

not consider spatial data of the pixels; instead, it provides a distribution of color or intensity in the form of bins.

Many of the shot boundary detection techniques, were based on the color histogram. The color histogram based methods, use different color spaces like RGB, HSV etc. Zhang et al. [95], calculates the difference of color histogram to detect the shot boundary.

However, there are few problems with this approach, even if two different images having a similar color distribution will end up showing a higher similarity score. Some improvement was suggested on this approach where an image is divided into smaller blocks, and a histogram is computed for the block, resulting in localized features The final histogram is a combination of all the histograms for individual blocks. Such an approach is presented in [68].

Pass et al. [72], compute color coherence vector to compute the difference between two images, they add the spatial information in addition to the color histogram. But adding the spatial information also adds sensitivity to any motion. Adcock et al. [16], used color correlograms to implement video search. Amir et al. [17], used color moments based approach in their work.

3.1.1.3 Texture-Based Methods. Texture-based features are not based on color or intensity information but contain information of the surface and the neighbouring information of the object Amir et al. [17] used co-occurrence matrix and Tamura features in their work. In the work proposed by Hauptmann et al. [48], authors used Gabor wavelet filters. The mean and the variance of the outputs are combined to create a final texture feature vector. Some of the research has been presented by Li et al. [54] Discrete Wavelet Transform is used to determine the shot boundary. The advantage of using wavelet feature descriptor is that it gives the frequency information related to the image and is fast concerning computational time.

3.1.1.4 Compression-Based Methods. Little et al. [58], proposed differences in the sizes of JPEG compressed frames was used to detect the shot boundary. In one of the other work [19], authors devised a new mechanism of determining shot boundary by computing the discrete cosine transform (DCT) coefficients of a compressed frame to determine the similarity between the frames.

3.1.1.5 Edge Based Methods. Totterdell et al. [85], proposed technique, based on the changes in the edges between two frames, this approach is insensitive towards illumination changes and robust to the motion. The shot boundary was declared by calculating the ratio of change of incoming and outgoing edges [56]. Hauptmann et al. [48], proposed to use the edge histogram descriptor (EHD) to capture the spatial distribution of edges. EHD is calculated by considering the number of pixels that form edges. The features can be computed locally by splitting the image into blocks. Foley et al. [44] and Cooke et al. [34] proposed to divide the image into 4x4 blocks, and an edge histogram was computed. Such features work best when the shape is dominant in the video.

3.1.1.6 Local Features. Scale Invariant Feature Transform (SIFT) developed by David Lowe [59] is one of the methods that detect robust features in images for the Shot detection. Li et al. [55], presented their approach to detect the shot boundary based on SIFT features. Inspired by SIFT algorithm, another feature called Speeded Up Robust Features (SURF) proposed by Bay et al. [22], has running time lesser than SIFT. Baber et al. [20], proposed to use the SURF features to determine shot boundary and they show that the features based on SURF, detect not only abrupt cuts but also fade-ins and fade-outs efficiently.

3.1.1.7 Motion-Based Methods. Motion is a useful feature to determine shot change in the video. Motion features are classified into two types. The first one

represents camera motion like zooming or panning. Ueda et al. [88] and Zhang et al. [95], devised a block matching algorithm to detect the zooming a panning.

The second one is the motion of an object which can be estimated by a motion vector. Ma and Zhang [61], transformed the motion vector to many directional slices according to the energy of the motion. A set of moments are computed on each slice and is transformed into a multidimensional vector called motion texture. This vector is used to determine shot boundary detection.

3.1.2 Similarity Functions

The similarity between two feature vectors is computed using p-Norm described as follows:

$$||x||_p = \left(\sum_{i=1}^{n} x_i^p \right)^{(1/p)} \tag{3.1}$$

where, p=2 for *Euclidean* distance.

Another metric used commonly is the *chi-square* given as

$$\tilde{\chi}^2(i,j) = \sum_{k=1}^{M} \frac{(I_i(k) - I_j(k))^2}{I_i(k)} \tag{3.2}$$

where, I_i and I_j are the i^{th} and j^{th} frames.

3.1.3 Shot Boundary Decision

In the feature selection step of shot boundary detection, the feature vectors are computed using one of the methods described earlier. Once we have the feature vector, the next step is to declare a shot boundary. Shot boundary decision can be taken by different approaches as follows:

3.1.3.1 Fixed Threshold. A global threshold is set for the cut detection. If the distance between two feature vectors is higher than the set value, then the cut is

declared [28].

$$\begin{cases} Dist_i \leq Tc, \text{noise} \\ Dist_i > Tc, \text{cut} \end{cases}$$

where, Tc is the threshold that needs to be selected. Choosing a correct threshold is the key to detect the cut. However, a single threshold does not work for every kind of video.

3.1.3.2 Adaptive Threshold. A threshold that varies with every video could be set to address the problem of the fixed threshold. To select a threshold, we must consider the distribution of inter-frame differences (or similarity) obtained and adjust the threshold accordingly [95]. Some statistics (like mean, standard deviation, etc.) could be applied to the differences within a temporal window [93].

3.1.3.3 Machine Learning based approaches: SVM and KNN. Machine learning approaches train a classifier that classifies a given shot into one of the two categories namely "shot change" or "no shot change." Such approaches often require training data to train the classifier. The training data is often prepared manually and labelled as one of the classes "shot change" or "no shot change." Lack of training data available on educational videos makes it difficult to use machine learning approaches.

3.2 A Comparison of State-of-the-Art Image Descriptors

The first step of transition detection is to define feature descriptors for the educational videos. The following descriptors were considered for the study based on their ability to detect texture features. In this section, the performance of Histogram of Oriented Gradients (HOG), Color Moments, Edge Change Ratio (ECR), Fast Fourier

Transform (FFT), Scale Invariant Feature Transform (SIFT) and Haar Wavelet descriptors are compared.

3.2.1 Histogram of Oriented Gradients

The main idea behind *Histogram of Oriented Gradients (HOG)* [36] is that object shape can be well described by the distribution of the gradient or edge directions without accurately knowing the position of the gradient. The image is divided into small regions called cells, and a histogram is calculated for every cell. These cells can be rectangular or radial in shape, and a gradient is computed for each pixel inside the cell. Each pixel casts a weighted vote for an orientation-based histogram channel. A separate gradient is computed for each color channel, and one with the largest norm is considered as the pixel's gradient vector. For each cell, a 1D histogram of gradient direction is computed by applying 1-D centered, point discrete derivative mask in horizontal and vertical directions. The histogram consists of 9 bins from 0 to 180°.

These entries are then combined and contrast-normalized. For contrast-normalization, the entries are accumulated to larger blocks, and all the cells in the block are normalized. The final descriptor is a vector composed of all the normalized cell responses from every block in the detection window.

3.2.2 Color Moments

Educational video consists of recording of power-point presentation presented by an instructor during the lecture. The power point has same template for all the slides. Thus, all the slides have similar background with only change in the content (text and images). Thus, we focus on a window centered in the video frame, as the change occurs usually in the center of any slide. We determine the window size experimentally. We further divide it into 8X8 blocks. By using this approach, we can record the changes locally.

For each block we calculate the 10 moments for R, G, and B respectively. The first moment is the mean of histogram and is defined as the mean.

$$m_1 = \frac{1}{N} \times \sum_i Hist(i) \tag{3.3}$$

Second moment is variance, third is skewness and fourth is kurtosis. Moment five to ten are the central moments. From 2 to 10, moments are computed as follows.

$$m_k = \frac{1}{N} \times \sum_i (Hist(i) - m_1)^k \tag{3.4}$$

where, k =2...10, N is the total number of pixels, Hist(i) is the histogram of the i^{th} block.

3.2.3 Edge Change Ratio (ECR)

The edge change Edge change Ratio (ECR) is based on the principle that the change in contents appears near the shot boundary. In ECR, the image is converted to edge image.

$$ECR = ECR(n,k) = max(\frac{Xin}{\sigma_n}, \frac{Xout_{nk}}{\sigma_{nk}}). \tag{3.5}$$

Where, Xin is the number of pixels in frame n,

Xout is the number of exiting pixels in the previous frame *n-1*, σ_n and σ_{n-1} are the number of edge pixels respectively in the frames n and $n-1$.

The edges are calculated by Canny detector [25].

3.2.4 Fast Fourier Transform (FFT)

Fourier transform converts the image in the frequency domain where an image is represented as real and imaginary components. The number of frequencies involved in an image corresponds to the number of pixels of the image. Fast Fourier Transform (FFT) provides a fast way of computing the 2D transform by calculating

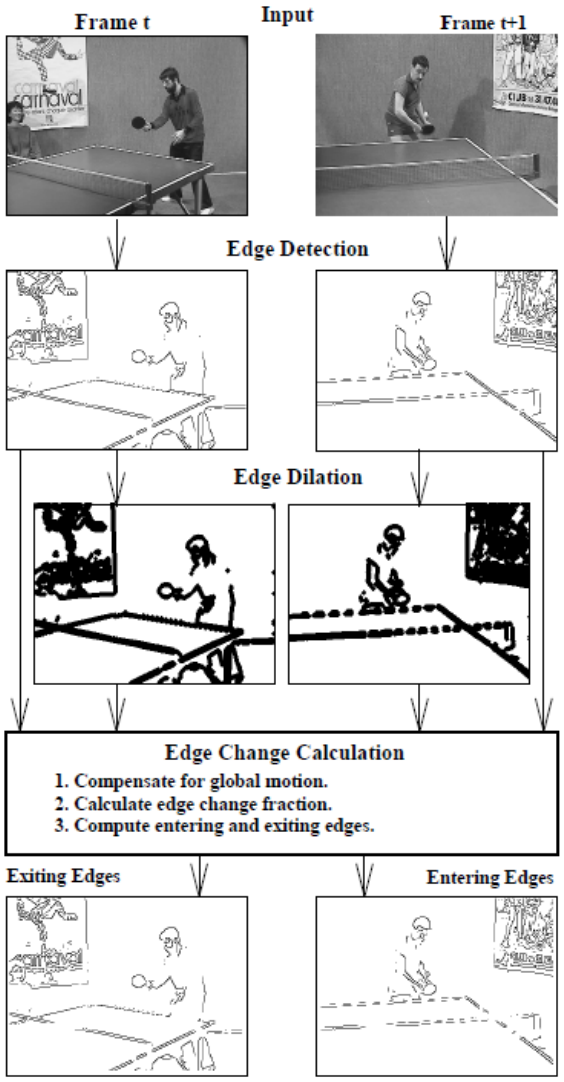


Figure 3.2 Edge Change Ratio(ECR), is calculated from the number of exiting edges in frame t and entering edges in frame t+1.

Source: A Feature-based Algorithm for Detecting and Classifying Scene Breaks [94].

the components at one time in the horizontal direction and then in the vertical direction. In FFT, the energy is concentrated in a circle at the center (Figure 3.3). The magnitude of the FFT component is considered and the feature vector generated by calculating the sum of the magnitude for each angle varying from 0 to 360°.

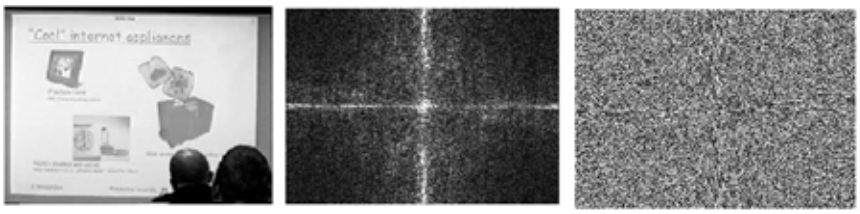


Figure 3.3 FFT 2D (original(left), magnitude (center), phase(right)).

3.2.5 Scale Invariant Feature Transform (SIFT)

SIFT produces key points in an image regardless of the scale change. There are different stages of detection of SIFT features. The scale space $L(x, y, \sigma)$ of an image $I(x, y)$ is defined as:

$$L(x, y, \sigma) = G(x, y, \sigma) \times I(x, y) \tag{3.6}$$

where, x,y are the pixel coordinates of image I, σ is the scale, $G(x, y, \sigma)$ is the Gaussian kernel

$$G(x, y, \sigma) = \frac{1}{(2\pi\sigma^2)} e^{(\frac{-(x^2+y^2)}{2\sigma^2})} \tag{3.7}$$

SIFT uses Difference of Gaussian (DOG) to detect the keypoints.

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \times I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{3.8}$$

where, $L(x, y, k\sigma)$ is the convolution of the original image I(x,y) with the Gaussian blur $G(x, y, k\sigma)$ at scale $k\sigma$.

The DOGs are computed by Gaussian smoothing the image at two different scales (Figure 3.5) σ, and computing the difference. The process is repeated for different octaves by reducing the resolution of the image by half for each octave. After the DOG is determined, the extrema are found by comparing one pixel in an image with its eight neighbors as well as nine pixels in next scale and nine pixels in previous scales. If this pixel is local extrema that is, if it is larger or smaller than all

Figure 3.4 SIFT Key point Matching between two frames.

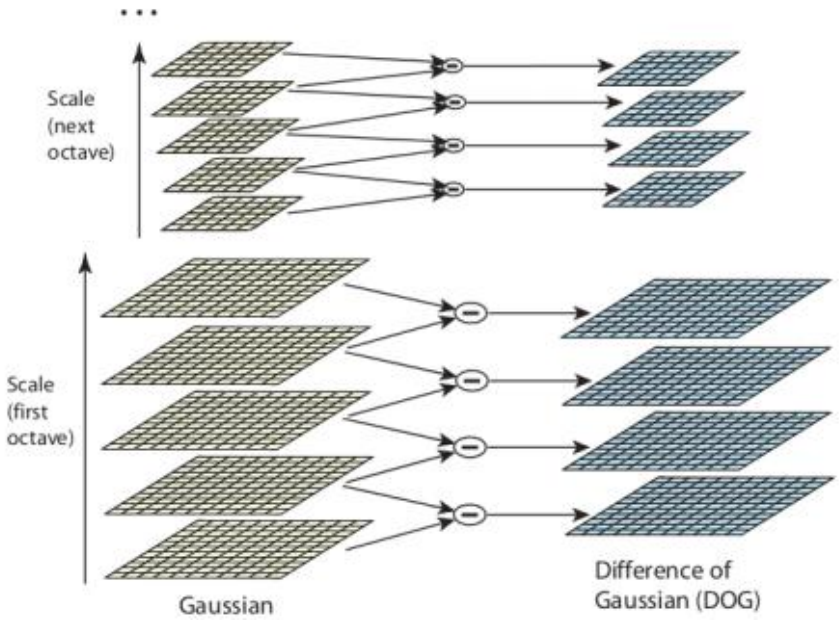


Figure 3.5 SIFT Algorithm. For each octave of scale space the image is convolved with Gaussians to produce the scale spaces (on left), these adjacent Gaussian images are subtracted to produce difference of gaussian images (on right), Gaussian image is down sampled by 2 and process is repeated.

Source: David Lowe [59]

these neighbors, then it is chosen as a potential key point. The next step is to localize the key points. If the intensity at the extrema is less than some peak threshold (0.03 as described in [59]), it is rejected. In this step, all the edge key points and the ones having low contrast is rejected. To generate stable keypoints, it is not enough to reject the keypoints with low contrast; the difference of Gaussian produces strong

edge responses. To eliminate the poor peaks in the difference of Gaussian function 2×2 Hessian matrix is used.

Once the key points are estimated, the next step is to assign the orientation, which gives stability towards image rotation invariance. An Orientation histogram is computed with 36 bins covering 360 degrees. The peaks in the orientation histogram show the dominant directions. The highest peak and any peak above 80% of the highest peaks are considered as orientations.

Next a descriptor is created by taking an area of 16×16 pixels around the key point. This area is divided into 4×4 sub-block. For each sub-block eight bin orientation histogram is calculated, so the final descriptor has 128 values. Best match for each keypoint is chosen as the nearest neighbor. As proposed in [59], instead of keeping a global threshold the ratio of the distance between closest and second closest neighbor is computed. This measure works better as the correct matches will have neighbors relatively closer than the incorrect ones. All the matches that have a ratio greater than 0.8 are discarded (Figure 3.4). Number of matches is not a good measure of similarity; the distance is given by:

$$dist = 1 - \frac{(Numberofmatchedkeypoints)}{(Totalnumberofkeypoints)} \tag{3.9}$$

3.2.6 Haar Wavelet

Wavelet is helpful in decomposing the image into sub-bands. It has an advantage over the Fourier Transform such that it carries not only frequency information but also the location information (temporal information). Discrete Wavelet Transform (DWT) is used to reduce the computations used in Continuous Wavelet Transform (CWT). It consists of high pass and low pass filter. We chose *Haar wavelet* as it possesses many qualities like good image features and fast processing.

After performing a 2D DWT, an image is decomposed into four sub-bands which are a quarter the size of original image. These four sub-bands are low frequency (denoted by LL) which is the first down-sampled approximation of original image, vertical detail (LH) which is the high frequency in the vertical direction (y-axis). HL is the high frequency in the horizontal direction (x-axis), and HH is the diagonal high frequency, which is directional difference diagonally. The LL band can be further decomposed into four bands producing quarter size output, with LL, LH, HL and HH bands. The LL band contains the major image energy and features whereas LH, HL and HH bands consist of vertical edge information.

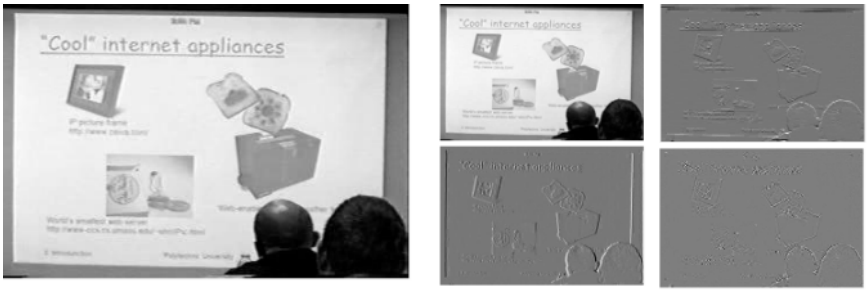


Figure 3.6 Wavelet decomposition, when wavelet transform is applied on image, the image is decomposed into various bands as seen (original image, LL, LH, HL and HH bands).

A similar approach used by Li et al. [54] is used to divide the image into n×n blocks, for each block DWT is computed. Each of the blocks has four coefficients, one for each of the sub-bands. All the sub-bands are used to calculate our feature vector to get the energy and edge differences. The first feature vector is the energy feature vector which is computed as follows:

$$E_f = (C_1, C_2 \ldots\ldots C_M) \tag{3.10}$$

$$d_f = E_f - E_{f+1} \tag{3.11}$$

$$D_{LL} = \sum_{k=1}^{M} d_f(k) \tag{3.12}$$

where, $C_1 \ldots C_{\mathrm{M}}$ are the coefficients for LL band for each block, M represents the total number of blocks, E_{f} is the Energy for Frame f, $E_{\mathrm{f+1}}$ is energy for next frame.

Similarly, the edge difference $\mathrm{D_{LH}}$, $\mathrm{D_{HL}}$ and $\mathrm{D_{HH}}$ is computed. Finally, the feature vector consists of four values obtained from each of the sub-bands.

After computing feature difference for frames using equation 3.11, the threshold is identified as the mean value for each of the sub-band. If the value is greater than its corresponding sub-band mean for each of the four sub-bands, a potential shot change is declared.

3.3 Video Dataset and Comparison Results

Key-frames are picked and Euclidean distance is calculated between the feature vectors of consecutive key-frames. The most natural approach to selecting key-frames is to choose a frame at a fixed time interval. A smaller time interval will pick more key-frames which will increase the accuracy but also the processing time, whereas transitions may be missed with substantial time intervals. We need to know the transitions to select an optimum value of m (the minimum time interval between two consecutive transitions).

For the videos recorded with regular cameras, the slides do not necessarily appear in all the frames and may not always occupy the entire video frame. The lecture videos can sometimes contain frames that are not slides (e.g. the narrator frames, audience, web page etc.). A classifier proposed by Dorai et al. [39], is used to classify the frames into slides and non-slides. Only slide frames are considered for the experiments.

Fourteen different lecture videos of varying quality were used for the experiments (Table 3.1), ranging from 30 minutes to 4 hours. The videos VD1 to VD10 were recorded with the regular cameras and the videos VD2, VD7 and VD8 were of lower quality. The videos VD1 to VD10 were full-screen videos.

Videos VD11 and VD12 were recorded using Camtasia and contained metadata. However, the instructor browsed back and forth in the slide presentation. The associated metadata file included incorrect slide numbers and inaccurate slide transitions. Also, VD11 and VD12 had slide covering partial part of the frame ranging from 70% to 80% of the frame. Both videos were provided by two different instructors.

Videos VD13 and VD14 were the most challenging lecture videos. Both were of poor quality with various problems, such as inadequate illumination, zoom effect and occlusion. Video VD14, had both the presenter and the slide in some of the frames, with the slide covering only 40% part of the screen and text appears line by line.

Table 3.1 Video Datasets

Video	Transitions	Duration	Metadata available?	Quality	Size
VD1	34	01:11:04	No	Fair	Full screen
VD2	19	00:38:33	No	Fair (Blurred characters, noisy)	Full screen
VD3	23	00:37:25	No	Good	Full screen
VD4	23	00:46:30	No	Good	Full screen
VD5	28	01:03:09	No	Good	Full screen
VD6	15	00:36:41	No	Good	Full screen
VD7	21	00:48:30	No	Fair, Blurred characters, noisy	Full screen
VD8	23	00:42:47	No	Poor, Very noisy and blurred characters	Full screen
VD9	19	00:40:22	No	Good	Full screen
VD10	30	00:47:26	No	Good	Full screen
VD11	70	01:48:45	Yes	Good	Partial screen (70%)
VD12	112	04:24:02	Yes	Good	Partial screen (80%)
VD13	14	00:22:05	No	Poor, very noisy	Partial screen (80%)
VD14	14	00:18:00	No	Poor, gradual slides	Partial screen (40%)

Table 3.2 illustrates the results that we obtained for shot boundary detection after comparing the above image descriptor techniques. For each of the techniques we selected an automatic threshold by using Dugad factor [41] which is given as follows:

$$T = \mu + t_f \times (\sqrt{\sigma}) \tag{3.13}$$

where, μ is the mean of distance calculated between two keyframes, t_f is the threshold factor, σ is the standard deviation of the distance.

If the distance calculated is greater than the threshold, then transition is declared. For our experiments, the threshold factor is set (t_f) as 2. The results show that features selected using HOG, Color moments and SIFT are among the best. Wavelet method has low recall rates. ECR is very sensitive to effects and quality.

3.3.1 Slide Matching

The slide matching phase ensures that the mapping between the slides and the video segments is in a correct order. In some cases, meta-data associated with screen capture videos had missing transitions. The slide matching phase is essential to align slides and videos correctly. Sometimes, the presenter can hide some slides during the presentation, these slides do not appear in the recording, while they exist in the presentation.

The slide matching phase consists of matching a slide found in a video frame with the actual power-point presentation slide converted into an image, and the extracted features (HOG) are compared with the features obtained (HOG) from the slides extracted from the video frames.

As shown in table 3.3, the slide matching phase corrected the transitions and improved the accuracy. The accuracy has improved in all videos except VD4, VD8 and VD14, for which, the quality was the major issue.

Table 3.2 Comparative Results for Transition Detection (Precision and Recall)

Video	Transitions	HOG	Wavelet	SIFT	Color Moments	FFT	ECR
VD1	34	94.1 100	94.1 100	91.4 100	97 100	88.9 100	97 100
VD2	19	100 100	100 100	100 100	100 100	100 100	100 100
VD3	23	100 100	100 60.9	100 87	100 100	100 100	100 100
VD4	23	100 100	100 100	100 100	100 100	100 100	100 100
VD5	28	100 92.9	100 57.1	100 96.4	100 96.4	96.4 96.4	14.3 3.6
VD6	15	100 100	100 100	100 100	100 100	93.8 100	100 86.7
VD7	21	100 100	100 100	100 100	100 100	87.5 100	100 76.2
VD8	23	95.5 100	95.5 100	100 100	95.5 100	80.8 100	95.23 95.23
VD9	19	100 100	100 100	100 100	100 100	100 100	100 100
VD10	30	100 100	100 90	100 100	96.8 100	100 100	100 93.3
VD11	70	100 98.6	30.8 5.6	100 98.6	70 98.6	65.4 98.6	56.3 25.4
VD12	112	99.03 96.4	41.2 12.6	63 96.39	45.25 94.59	41.6 98.2	0 0
VD13	14	84.16 76.92	63.63 50	38.9 100	76.92 71.2	66.6 71.4	33.33 7.14
VD14	14	63.3 100	0 0	60 85.7	80 85.7	42 57.14	64.7 73.3

Table 3.3 Slide Matching Results

Video	*Transitions*	*Precision*
VD1	34	100
VD2	19	95
VD3	23	100
VD4	23	79.6
VD5	28	100
VD6	15	100
VD7	21	95.45
VD8	23	82.60
VD9	19	100
VD10	30	100
VD11	70	100
VD12	112	90.09
VD13	18	93.33
VD14	11	81.81

CHAPTER 4

ANNOTATING CLASSROOM VIDEOS WITH SLIDE LOCALIZATION

For classroom videos like Cases II and III, it is necessary that the slide is extracted before the transition detection phase. This helps to get rid of false cases, like motion of audience members, the speaker, etc. The rest of the steps for classroom videos are same as described in screen capture videos.

Slide localization is a technique of detecting and extracting slide in the video frames. This chapter focuses on different images in color space and proposes a well-suited algorithm for the scenario. We discuss various transformation techniques that are best according to color distribution between DCT, marginal and grayscale transformation. Many images are not color predominant, and such images can be represented effectively in less than two dimensions by transforming RGB space to DCT (dimension-1 and dimension-2), marginal space and grayscale, which merges all information on one dimension.

Segmentation techniques have been always an area of interest for researchers. Various types of segmentation techniques exist in literature. Common techniques include thresholding and edge detection-based methods. In this work, K-means technique is used. We evaluate the segmentation results of K means clustering on DCT, Marginal and Grayscale transformations. DCT yields results that are close to real image. If the image has color distribution limited to one dimension, marginal and grayscale are more suitable. We also compare the results of segmentation with ground-truth and evaluate the results with similarity measures.

Segmentation of educational video frames, poses several challenges. The images captured in a lecture video sequence have problems regarding the conditions in which video is shot, and generally, the quality is not very good, depending on various factors

discussed earlier. Localization of slides in such video frames is extremely difficult in such scenarios. Since the color distribution is usually limited to a single dimension in case of lecture videos, we focus only on grayscale and marginal space and evaluate the results. According to the evaluation results, we note that marginal performs slightly better than grayscale. Finally, we discuss the localization of the slide in a video frame. We show that after detecting different regions in marginal space, we can localize the slide efficiently by using simple heuristics.

4.1 Image Transformation Techniques

The best-known representation for a color image is the RGB space composed of 3 dimensions R, G and B. As we can see from the color distributions (Figure 4.1) that image (ia) has more scattered distribution (ib), which means it is more color predominant. Most of the images are not so color predominant for example image iia and iiia. For such images it is useful if we transform the image from RGB to DCT (which uses 2 dimensions), Marginal (1 dimension) and grayscale (1 dimension) as we do not need all the 3 dimensions in this case and we can make use of k means based on grayscale histogram approach to segment color images effectively. The suitability of each of these transformations is dependent on the distribution of color in an image (Figure 4.1). We explain each of the transformation technique in detail.

4.1.1 DCT Transform

Translation of RGB space to DCT is given by the following equation

Wm(k) =

$$\begin{cases} \sqrt{\frac{1}{\sqrt{3}}} \text{ for m=1 and k=1,2,3} \\ \sqrt{\frac{2}{3}} cos((2k-1)(m-1)\frac{\pi}{6}, \text{for m = 2,3; k = 1,2,3} \end{cases}$$

This space preserves the non-correlation of data and preservation of total energy.

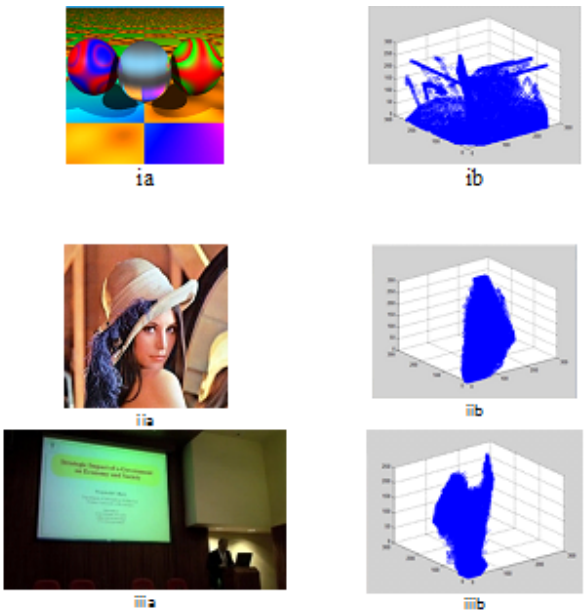


Figure 4.1 Color Distribution(3D) for 3 dimensions R,G and B, as represented above a) Original Image b) 3D color distribution of the image corresponding to R,G,B dimensions.

Figure 4.2 DCT Transform on image: a) Original image, b) DCT dimension 1 c) DCT dimension 2 d) DCT dimension 3.

4.1.2 Grayscale

A grayscale image is an image in which the value of each pixel carries only intensity information. Images of this sort, also known as black-and-white, are composed of shades of gray, varying from black at the weakest intensity (0) to white at the strongest (255).

Figure 4.3 Grayscale conversion of images. Here the image loses the color information, but the intensity information is used.

4.1.3 Marginal

Marginal space is useful for images where color information is not that predominant, or the color distribution is limited to one dimension. HSV space provides a better de-correlation of information in the visual sense. In this space, color information can be reduced to a composite monochrome image by Carron's [27] criterion which digitally merges Hue (I_H), Saturation (I_S) and Value (I_V), information into a single magnitude M defined by:

$$M = \alpha(I_S)I_H + (1 - \alpha(I_S))I_v \tag{4.1}$$

$$\alpha(I_S) = 1/\pi[\tan(\beta(I_S - S_0))] \tag{4.2}$$

where, $S_0(0 \leq S_0 \leq 255)$ defines a mean relevance level of the hue related to a saturation level, and $\beta(0.05 \leq \beta \leq 0.5)$ used to tone the mix. Thus segmentation techniques developed for gray-scale images can be used for this case.

For lower values of S_0 there is a clear distinction between different colors in the image and higher value of S_0 is close to the grayscale value. The value of S_0 and β vary for every image. For the experiments, the value of β was chosen as 0.05 experimentally and S_0 was chosen as the median value of the distribution that varies for each image.

Figure 4.4 Marginal Image. The figure shows a comparison between grayscale and marginal image. While marginal is also 2D image, it carries more color information as compared to grayscale. a) Original image b) Grayscale c) Marginal(S_0=74, β= 0.05) d) Marginal(S_0=255, β =0.05).

4.2 Segmentation

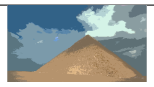
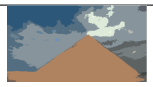
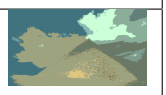
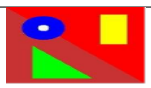
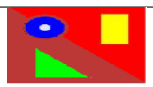
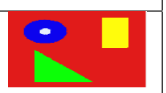
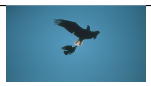
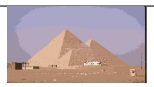
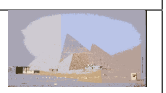
In this section the segmentation results on DCT, marginal and grayscale spaces are compared. To segment and localize the slides in a lecture video, we first need to analyze these three techniques on general images. The idea here is to study segmentation of these transformations to find their suitability on different types of images. We use K-means technique to segment the image into different clusters. K-means is a classical technique widely popular in image segmentation.

4.2.1 K-means Clustering

We use the grayscale histogram approach to form the clusters. In general, we observe that choosing random centroids does not yield same result for every run, so we fix the initial centroid using Tsai's moment-preserving method [86] with multiple thresholds. For DCT, K-means clustering is performed on dimension 1 and dimension 2 separately and then both the regions are merged. We use Berkeley dataset [63] and ground truth for general comparison. For some images we compute the ground truth ourselves. The visual comparison results of K-means are formulated in Table A.1.

We use the grayscale histogram approach to form the clusters. In general, we observe that choosing random centroids does not yield same result for every run, so we fix the initial centroid using Tsai's moment-preserving method [86] with multiple

Table 4.1 Mean Color Image Obtained after Clustering in DCT, Marginal, and Grayscale Space

	Original Image	*Ground Truth*	*DCT*	*Marginal*	*Grayscale*
(1)					
(2)					
(3)					
(4)					
(5)					
(6)					
(7)					
(8)					
(9)					
(10)					

thresholds. For DCT, K-means clustering is performed on dimension 1 and dimension 2 separately and then both the regions are merged. We use Berkeley dataset [63] and ground truth for general comparison. For some images we compute the ground truth ourselves. The visual comparison results of K-means are formulated in Table A.1.

4.3 Similarity Measures

To compare different segmentation results from DCT, marginal and grayscale, we make use of similarity measures. First, we compute a confusion matrix between the ground truth and the regions obtained in segmentation and then we compute the following measures.

4.3.1 Jaccard Index

We calculate the Jaccard Index similarity measure defined as

$$J = \frac{A \cap B}{A \cup B} \tag{4.3}$$

where, A and B are ground truth and segmented image respectively. "$\cap$" is the intersection between the two sets and "$\cup$" is the union of two sets. Jaccard Index is calculated for each region and then overall similarity is calculated as mentioned by Busin et al. [24].

4.3.2 F-measure

We calculate F-measure as follows

$$F\text{-}meas(\beta) = \frac{(1+\beta^2)(Precision \times Recall)}{\beta^2 \times (Precision + Recall)} \tag{4.4}$$

where, β gives β time importance to recall than precision. Precision is given more weight than recall as discussed by Achanta et al. [13].

As seen from Table A.1. Both Jaccard and F-measure give a consistent result for all the images. We can see that most of the color distribution of images is limited to one dimension except image 2, that is due to the over-segmentation caused by labels marked in benchmark image, which do not consider other details of the image. For the first two images, ground truth is created manually. For most of the results, DCT shows clusters close to the real image as it takes account of every detail in the image. A better merging approach is needed to avoid over-segmentation.

Segmentation in marginal space omits some details, but still manages to capture the necessary details; it performs well on average with benchmarks, and generally performs better where the color distribution of the image is not scattered. We also analyze whether the S_0 we obtain is ideal or not. In most cases, the median value is

close to ideal value. in images 1, 118035, 302003 and 388016 the median S_0 value is not ideal. Hence, we can improve the results by tuning the S_0 value.

Grayscale does not perform well when there are subtle color changes, which is evident from the image 5 of Table A.1. If the color distribution is limited to one dimension and there is enough contrast between the objects, grayscale performs better.

The running times of the k-means on the three spaces are compared, for some images marginal performs faster than grayscale, and in others, grayscale is faster. DCT takes most time as K-means is performed on two dimensions separately.

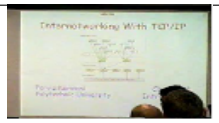
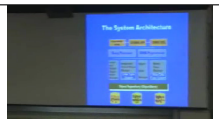
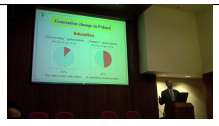
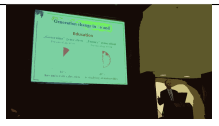
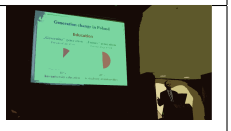
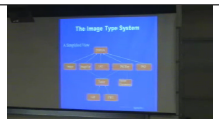
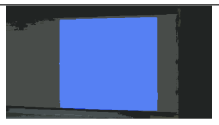
4.4 Slide Localization

In the previous section, general images were used, and we can conclude that for the images which are not color predominant marginal and grayscale transformation is enough to represent the image. In this section, our focus is on educational videos. Since the lecture video is not so color predominant and the color distribution is limited to one dimension, instead of DCT (which uses two dimensions for analysis), grayscale and marginal space are considered. For the lecture videos, a visual comparison is presented, since the ground truths of these images are not available.

For localizing the slide, the value of S_0 is chosen experimentally. Marginal space gives the flexibility to tune the parameters. From the visual comparison in Table 4.2, it can be seen that grayscale and marginal yield similar results. For some images grayscale loses some part of the slide whereas marginal can detect it.

Marginal space performs better than grayscale in some cases, which is evident from Table 4.2. Hence, segmentation results of marginal space are used for slide localization.

Table 4.2 Mean Color Image Obtained after Clustering in Marginal and Grayscale Space

	Original Image	*Marginal*	*Grayscale*
(i)			
(ii)			
(iii)			
(iv)			
(v)			
(vi)			

4.4.1 Heuristics for Slide localization

Two heuristics are used to detect the slide regions from the regions obtained by k-means performed on marginal space. The heuristics are proposed, according to the standard observations in a recording of lecture video (Table 4.2).

Size: In a lecture recording, slide usually covers the significant part of the video frame. In practice, the slide covers at least (1/4) of dimensions of a video frame.

Luminance: In any presentation, the slide region is the most illuminant region than the surrounding.

Based on the results obtained in Marginal space in Table 4.2, the first heuristic is used to calculate the size of each region. An adaptive threshold is set, based on the

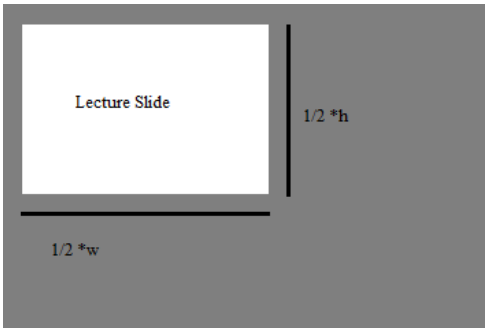


Figure 4.5 Heuristics for slide localization are based on size of the region after segmentation and the intensity of the region.

size of the image (1/4 * size of frame). The regions having a size larger than the threshold are identified as candidate regions.

Intensity of each region is computed using heuristic 2, and the best candidate is recognized as the slide region. There can be more heuristics associated with the slide, such as shape etc., but the two heuristics were enough to localize the slide for four different datasets.

Figure 4.6 Extracted slide, obtained after applying localization using above two heuristics and segmentation in marginal space.

4.4.2 Results

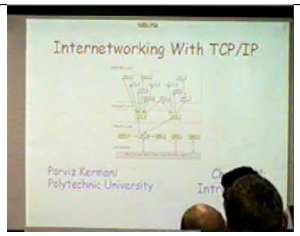
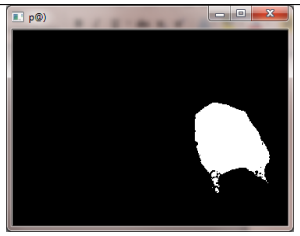
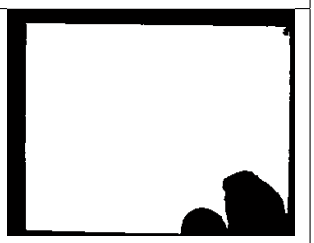
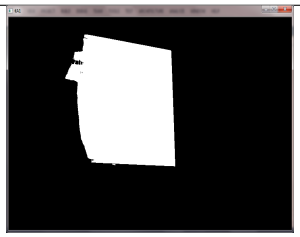
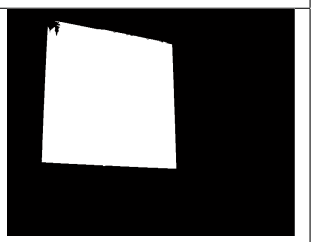
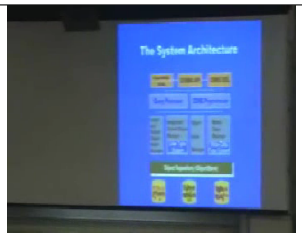
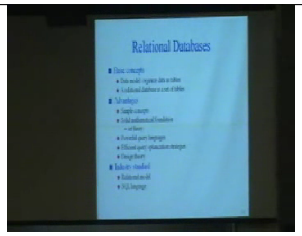
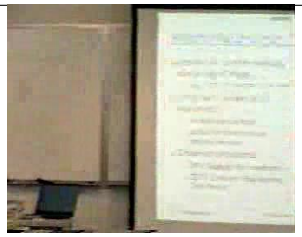
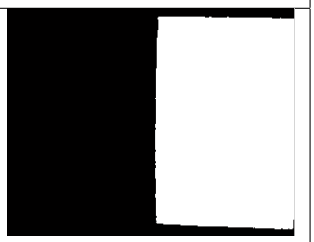
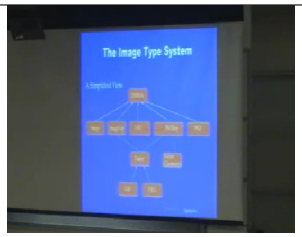
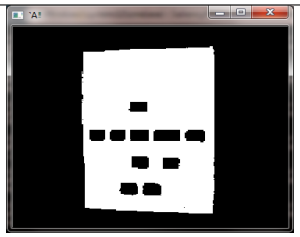
After applying localization algorithm on two classroom videos (VD13 and VD14) using HOG, it is observed that the precision was improved from 84% to 92.8% for and recall to 85% for VD13 and VD14 from 63.63 to 92.8% (precision). For SIFT the precision was improved from 38.9% to 43.75% for VD13 and for VD14 the precision improved from 60 to 68.4% Using Moments, the precision for VD13 improved from 76.92% to 92.3% and for VD14 from 80% to 85%. Best results were obtained for HOG among all the image descriptors as mentioned earlier.

4.4.3 Saliency vs Localization

An alternative way to detect an object is known as visual saliency. Saliency is closely related to what human find most interesting when they first look at an image. In a lecture recording, the most interesting object is the slide, as it is most illuminated. One of the works in this area is the saliency filters [73]. To find the saliency, the authors first segment the image using superpixels [14]. Once the image is segmented into superpixels, the abstraction phase removes any unwanted details to create a homogeneous distribution of pixels into different regions. For every region, uniqueness and spatial distribution is computed, and this contributes to the final salient region. We compare our method of slide localization to salient object detection. The results are tabulated based on the same input given to the two algorithms (Table 4.3).

From the results (Table 4.3), it can seen that for Case (i) a very small part of the slide is extracted when we use saliency whereas localization can retrieve the entire slide. Similarly, for Cases (ii) (iv) and (v) the saliency algorithm extracts only partial slide, whereas by using localization approach, the entire slide can be extracted. For Cases (iii) and (vi), the saliency results are similar to the localization approach. Overall the slide localization yields better results than the saliency for slide extraction.

Table 4.3 Saliency Vs Localization on Video frame for Slide extraction

	Original Image	*Slide extracted with Saliency*	*Slide extracted with Localization*
(i)			
(ii)			
(iii)			
(iv)			
(v)			
(vi)			

CHAPTER 5

INDEXING, RANKING AND UCS APPLICATION

Ultimate Course search (UCS) aims to automate the whole indexing process as much as possible. In UCS, image processing techniques are used to detect shot boundaries for the videos. The transitions are detected using HOG features. HOG features are more robust than the pixel-based and frame differencing methods, used in many of the works in mapping the slides and video. The HOG feature descriptors used to identify the transition, are accurate in detecting the transitions even with the slides that have same titles.

UCS is meant for students participating in classroom lectures. UCS provides all the media, such as textbooks, videos, and slides, on a single platform. The content is present in one place, with a search feature, so that users can quickly search through the lecture material without having to go through the entire content. Users can enter a keyword corresponding to the topic in which they are interested, and the results are displayed on the interface, this helps the user to prepare and study for the content in the most efficient way.

UCS has one more feature that makes it different from the other work: the ranking mechanism that considers the region of the appearance of the search keywords, this enables the users to get the most relevant result set for the topic they are interested.

To make the multimedia learning materials searchable by their learning content, we need to index them by their learning content. Separate indexes are generated for slides, videos and textbooks. The slides are indexed on the keywords extracted from PowerPoint presentation. The videos are mapped to the slides. The following subsections describe the steps that take place for the data annotation and indexing.

5.1 Indexing

5.1.1 Slides as a Roadmap to Learning Material Annotation

PowerPoint slides are a very common medium of teaching. They are carefully prepared by the instructor of the course who is often an expert in the area. We extract the text from the PowerPoint slides. We also extract structure-related information using Apache POI [18], a Java library for reading and writing files in Microsoft Office formats. The text in the slides is processed using classic text processing techniques like tokenizing and stemming.

Each word extracted using above process is compared against the course ontology [89], when available. In the simplest form, the ontology can just be a taxonomy provided by the course textbook index. As the back index of the textbook is provided by an expert, its keywords are likely to be used in learning material searches. Indexing only the keywords in the slides that appear in the textbook back index helps us reduce the number of keywords to be indexed.

For each keyword previously extracted from the slide text, we store its region-based information (e.g. slide title, subtitle, and text). This helps us at the ranking stage as we can assign different weights according to where the keyword appears in a slide. The slide index is composed of documents, where each slide is treated as an individual document. For each slide document, metadata such as presentation identifiers, slide numbers, and titles are also indexed. Since the PowerPoint presentation consists of a set of slides, the entire presentation is treated as a composite document that is also indexed. We create link between individual slide and the presentation which helps us to identify whether a slide is a part of a particular presentation.

The keywords are stored as inverted lists for indexing the learning material. The inverted lists help to quickly fetch the set of documents that contain a given term. The slides and the presentations that contain them are then ranked based

on the keyword location and term statistics such as frequency, term dictionary (all indexed terms and the number of documents containing these terms), term proximity (position of occurrence in the document), etc.

5.1.2 Video

Educational videos are a very popular teaching medium that capture not only the PowerPoint slides but also the instructor's explanation. The huge popularity of the videos is due to the e-learning concept, which is aimed at students who cannot attend the classroom lectures.

To make the videos searchable we need to index the videos. As discussed earlier the keywords are extracted directly from the video stream, which is a heavy process; instead, we make use of the keywords extracted from the slide and try to establish a relationship between the video and slides.

To link a slide to the part of video where the slide appears—i.e., finding a video segment that talk about a particular slide—we need to find the start and end time of the video segment associated with a particular slide so that users can view the corresponding explanation for the slide in the video. We use this information to build the video index. This mapping is called the slide-video index.

Often the lecture video also has certain frames which are non-slides: e.g., narrator frames or frames where the instructor explains a concept with the help of a command prompt or web browser. As a preprocessing step, we classify the frames into slides and non-slides and remove these frames from the set of candidate frames. This helps us pick the right keyframes and also reduce the number of false positives for transition.

For the lecture videos that are recorded with the help of lecture recording software such as Camtasia [83], we determine these transitions using the metadata file that comes along with the recording. For the videos that are recorded with software

but are missing the metadata or are recorded with regular camera, we use histograms of oriented gradients (HOG) [36] on the video frames to determine the transitions.

The slide-video index contains the information about slide number and presentation details they are associated with, along with start and end time of each video segment associated to that slide. Therefore, when users search for keywords in the search bar, the slide index is searched for the keyword and using the slide-video index, and the corresponding video will be linked and displayed.

5.1.3 Textbook

For learning any material thoroughly, we can get in-depth information from the textbook. We used electronic forms of textbooks given to us by respective authors. In order to look up any particular topic in a textbook, we normally look at the back index of the textbook, which provides us with the page number(s) on which this topic or term appears. We make use of same concept: we take the back index of the text-book in an electronic format [46]. We parse the keyword and page numbers and use it to create our textbook index. When a user searches for a keyword in the textbook interface, the keyword is searched in the textbook index and a list of matching terms is returned along with their page numbers. The indexes on slides, videos and textbooks have been implemented using Apache Lucene.

5.2 Keyword Appearance Region Prioritized Ranking

The classical document search based on term frequency and inverse document frequency (TF/IDF) alone will not yield the desirable result here as a high frequency of a term in a slide does not necessarily mean that the term is defined in that slide. We use the heuristic that if a keyword appears in the title then it is likely that the slide is about the term. We divided the region into two parts, the title and the body, which correspond to the slide title and slide text respectively.

To calculate the score for an individual slide (document), we use the TF/IDF measure and attach a weight depending on the region where the query term appears. If the query term appears in the title region, we give it a higher weight than the body. On the other hand, if the keyword appears in the body of the slide, it is given relatively lower weight. Given a query q composed of the terms t_1 ,…,t_n, the score of a document d (slide in our case) is computed as follows:

$$Score(q,d) = \sum_{t \in q} tf(t\ in\ d) \times idf(t) \times weight_{title} + \sum_{t \in q} tf(t\ in\ d) \times idf(t) \times weight_{body} \tag{5.1}$$

where, $weight_{title}$ is the weight applied if a query term t appears in the title, $weight_{body}$ is the weight applied when the term t appears in the body, *tf (t in d)* is the term frequency of the term in region (title or body) within the document, and *idf(t)* is the inverse document frequency given by $\log \frac{N}{df}$. Notice that $weight_{title} > weight_{body}$ and both values are greater than 1. The weight $weight_{title}$ is used to boost the score, when q query keyword appears in the title of the slide. The scores of individual query terms are added up to obtain the $Score(q,d)$ for a document d. The scores of individual query terms are added up to obtain the $Score(q,d)$ for a document d. The scores of individual slides in a presentation are aggregated to obtain the score of the presentation. The presentation score is also boosted with a weight ($weight_{presentation}$) a query term appears in the presentation title as the presence of the query terms in the presentation title may imply that entire presentation talks about this topic.

5.3 UCS Functionality Overview

UCS integrates learning materials from different media and allows them to be searched and viewed through a single interface. A student viewing a particular slide can also view its associated video segment and the corresponding textbook pages. The application is written in Java. We use Apache Tomcat as the web server and Apache Lucene as the search engine. UCS provides two types of searches: the first type is on

slides and videos combined, and second is on textbooks. When users provide keywords in the search bar of the slide and video interface, all the slides and corresponding video segments that match the keyword are displayed in the order of relevance. The top 20 results matching to a keyword are returned.

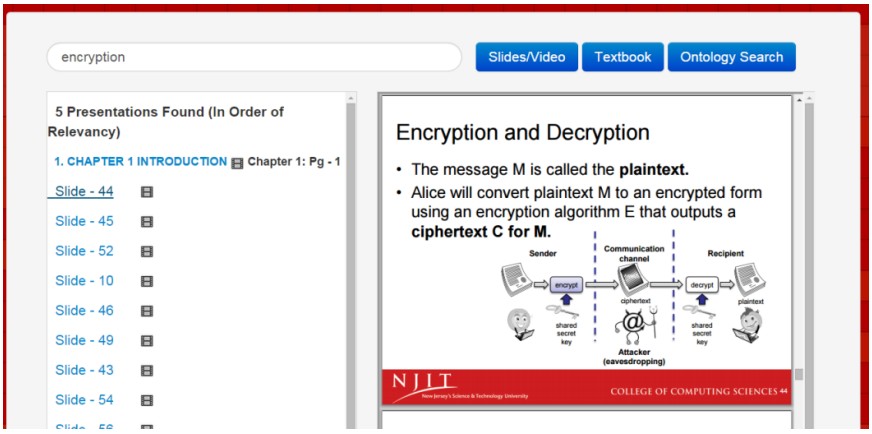


Figure 5.1 Slide/Video interface in UCS application with slide option selected.

When users type in keywords, they are presented with suggestions based on the keywords that are extracted from the PowerPoints. Upon clicking the slide/video button, internally Apache Lucene uses the slide index to fetch all the slides that contain the keyword. We prioritize the results according to the region-based scheme and the results are returned on the left side of the interface as shown in Figure 5.1. The results are displayed as a list of links where each link corresponds to an individual slide that contain the keyword. The links corresponding to slides from the same presentation are grouped together (i.e., presented consecutively). Upon selecting a link, a particular slide can be viewed in the display area on the right.

As shown in Figure 5.2, if a user wants to view the corresponding lecture video, then he or she can click on the video icon in the search results, and only the part of the video that is about this slide is played. The user does not have to go through the entire video to understand a topic. After a search, the user can also freely drag the cursor to play any part of the entire lecture. We use our video indexes to get

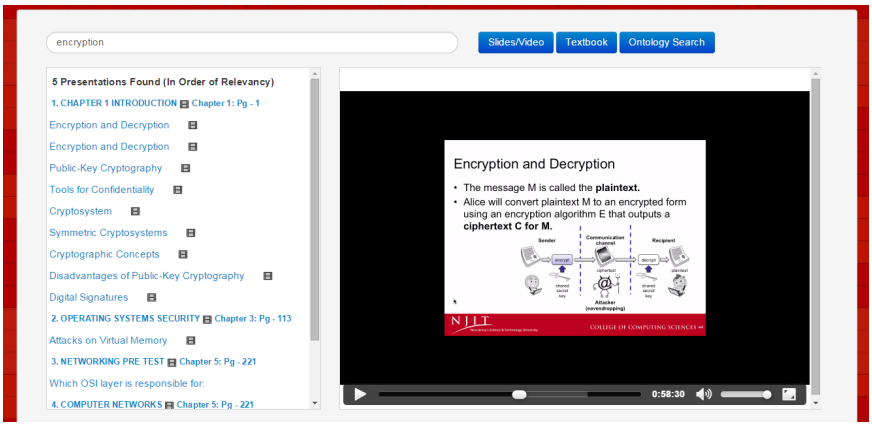


Figure 5.2 Slide/Video interface in UCS application with video option selected.

the timestamps of the beginning and end of the video segment and use html5 code to play only part of the video. UCS also provides logical connective operations such as "AND," "OR" and "NOT" to enhance the search. For example, if we search for "encryption NOT decryption," then only results for encryption will be displayed, and the results containing the keyword decryption will be omitted from the results. It is the same case for "AND," where if we search for encryption AND decryption, then the slides containing both these keywords are returned as top results.

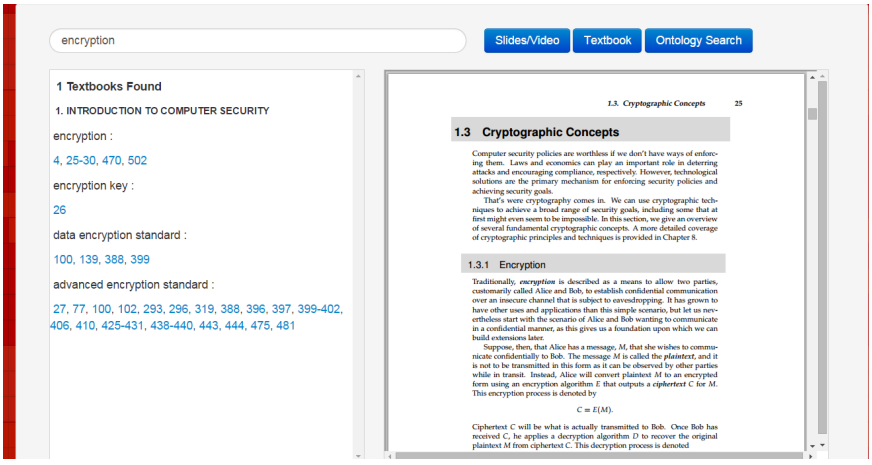


Figure 5.3 UCS application with textbook interface selected, when user looks for a search keyword, the results are presented as a list of page numbers. When user clicks on the page number result, that particular page number is displayed on the right side.

Figure 5.3 shows the textbook search interface that allows users to view the pages where the keyword appears. When the user types the keyword in the search bar, the result returns the list of the terms where this keyword appeared and the corresponding page numbers. On clicking the page number, users can view that particular page within the textbook with the keywords highlighted. This removes the need for users to go through the entire textbook and makes the textbook easy to navigate.

5.4 UCS Evaluation

Currently, UCS is in its beta version and was used in two courses at our university. Students utilizing the tool provided feedback to the research and development teams at the end of the semester. Feedback on UCS was requested in a questionnaire. Questions included ways to improve the user interface, how the students utilized the tool, how it affected their learning, and what positive aspects of UCS there were. Some students used UCS only when studying for tests or completing assignments, up to a few times during the semester, while others utilized UCS two or three times per week. The majority of students indicated they used the tool to study for tests, collaborate with peers, and review their notes. Typical comments included, "study for midterm," "to take better notes," and "look for terms." Users were asked "what effect did the tool have on your learning?" One student responded, "profound. Helped me understand the material more in depth." Students also stated that UCS "made me write detailed notes so I could do better in class," and "it made it a lot easier to look up information."

Students were also asked what they liked about the tool. The majority of responses centered around the usability and accuracy of the tool. Common comments included: "it is intuitive," "you can find the slides specifically with the key word", "quick search," and "fast search engine". Thus, students utilized the tool to solve

current problems in electronic course content. Students were able to use UCS to search for specific terms, as well as aid their studying and notetaking. Not having to search through all of the material "made it quicker."

While the majority of students found the tool easy to use, users provided feedback to the development team regarding improvements. Users requested larger window sizes for the textbook and videos, a longer period of time before timing out of the tool, the ability to highlight the search terms in the textbook, and more instructions on the use of the tool. Overall, the improvements requested focused on design rather than on the accuracy or ease of use, indicating the tool provided information in a timely way, and that the searches were accurate.

5.5 Conclusions

In this Chapter, Ultimate Course Search was presented, which provides not only a very simple-to-use interface but also a beneficial way to search various lecture material. For making the learning material search-able, we index the three most widely used lecture media: slide, video and textbook. We index the slides by identifying relevant keywords from the slide.

We show that without annotating the slides and videos we can effectively link the material by storing the transition of slides in a video. Our results show that finding the transitions automatically, along with matching with the original slides, helps us to identify better transitions.

UCS also offers to search in the textbook by indexing content from the back-index of the textbook along with page numbers. UCS thus integrates these three learning media into a single platform, which provides students with a way to search the material effectively and efficiently. The results presented to users after a keyword search are based on the region where the keyword appears, displaying the results in such a fashion brings the most important and relevant content on the top. Currently,

we are integrating the speech of the instructor, which will add many more keywords and make the interface for videos independent.

User research was conducted comparing security course students using UCS with students not using UCS. Both classes were taught by the same professor, using the same syllabus, assignments, and lectures. The attrition rate for the course utilizing UCS was 13% as compared to the attrition rate of 41% for the course without access to UCS (Renfro-Michel and Walo, in press). Students used the tool to study for their exam, watch lecture videos, search for specific terms and information, and to complete homework assignments and projects. Overall, the students using the tool found it to be user-friendly, fast and accurate, and stated that it helped them understand difficult course concepts.

CHAPTER 6

PERSONALIZED E-LEARNING SEARCH RESULTS: TAKING INTO ACCOUNT WHAT THE USER KNOWS

6.1 Introduction

Personalization, also known as customization, is the concept of presenting information that is relevant to the user: e.g., social media applications, a recommendation of television shows and movies, online advertisements based on previous searches, etc. Personalization may emphasize specific information related to a user; in other cases, the system can restrict or grant access to particular tools or interfaces depending on the user profile, or offer ease of access by remembering information about a user. Various tech companies such as Google, Facebook, Microsoft and Yahoo personalize user experience by building a profile that is based on the search history of the user. Amazon can provide customized offers to their customers from their purchase history. The advent of personal devices has popularized personalization. As a result, the content presented to the user has become concise and relevant.

In the case of e-learning systems, personalization can have different meanings ranging from adapting the content to the user learning preference or the knowledge level. Personalized learning starts with the learner. It means that learners have a say in their learning by taking responsibility for it. When they own and drive their learning, they are motivated to learn. Personalized learning tailors the environment to meet the learner's requirement.

Today, an increasing number of online learning resources are generated every day. As a result, users searching for a concept can get overwhelmed. The digital learning data can be leveraged in different ways to assist the user better. The standard way of learning and the concept of "one size fits all" is no longer the best way to learn, and there can be several ways to personalize e-learning.

6.1.1 Learning Preference

Learning preferences refer to a person's pattern of learning and preferences in processing and retrieving information [75], [29],[80]. In general, learning preferences can be categorized into the following:

Verbal/written: Learners who prefer learning by reading, and tend to remember and express the information by writing it down.

Aural/Auditory/Oral: These learners can learn better when they listen to explanations. Some auditory learners also prefer to read aloud to understand a concept.

Visual/Graphic: Visual learners are the ones who learn when they see something: e.g., figures, pictures, videos, etc. They also might prefer reading.

Active/Reflective: Active learners process information on the fly. They benefit from studying in groups. On the other hand, reflective learners spend time themselves thinking through the concept before joining in the group discussion.

6.1.2 Learning Concepts

When users want to learn a specific topic, they can be presented with in-depth suggestions or recommendations of concepts to better understand them. This information could be personalized based on user learning styles [37], [38], [49], [62]. Learning preferences can be broadly classified as verbal/written, visual and auditory learners. The user interface can be personalized based on individual users' learning preferences. User preferences can also be personalized based on user behavior and usage history. This can be done by tracking the user session and providing further recommendation based on users' behavior.

6.1.3 Personalized Learning in UCS

Students taking the same courses may have different knowledge levels due to previous courses. This is precisely the gap we would like to fill with the personalization

method we are proposing. This work is an extension of "Ultimate Course Search (UCS)" proposed by Rajgure et al. [74], designed for students in higher education. The learning materials in UCS are slide presentations, videos and textbooks and UCS provides an integrated and effective way to search these heterogeneous lecture materials. We have defined a course precedence graph that uses the course prerequisite information and the chapter precedence graph that defines guidelines for using course textbooks to define a precedence relationship for learning concepts. The user knowledge is based on the courses the user has already taken.

The rest of the chapter is organized as follows: Section 6.2describes some of the related work done in personalization. In section 6.3, we present the data that is used in our work, namely chapter precedence graphs, course precedence graphs, user knowledge graphs and query graphs. Section 6.4 presents query processing and the ranking mechanism used. Matching the query result and the user concept knowledge is presented in section 6.5. Section 6.6, provides some example queries to show the personalized result.

6.2 Related Work

6.2.1 Personalization Based on Learning Preferences

In the digital world, many efforts are made to cater the needs of user by studying and analyzing user data such as usage habits and preferences proposed by Brusilovsky et al. [23]. Several techniques have been proposed to mine users' data and offer personalized learning activities [45]. Chen et al., proposed a personalized course recommendation system based on Item Response Theory (PEL-IRT) [29] that considers both course material difficulty and learner ability to provide individual learning paths for learners. Learners' feedback responses are collected using feedback agents to improve the recommendations and the learner abilities are reevaluated. The study also proposes a collaborative voting approach for adjusting course material difficulty.

Intelligent Tutoring Systems proposed by Chen et al. [30] work on courses, such as geometry or physics education, as well as several Adaptive Educational Hypermedia (AEH) using both Adaptive Presentation to adapt the content of a page based on the student model, by inserting, changing and hiding specific fragments of text and Adaptive Navigation Support to adapt link presentation (and support the student's navigation) through annotation, sorting and hiding techniques [23].

Learners' most observed and modeled characteristic is their knowledge about the learning domain, assessed through quizzes or usage-based information. Some systems are based not only on modeling the students' knowledge, but also on their learning styles. By modeling the learner, learning systems can adapt content to the individual user's actual needs.

An intelligent agent called eTeacher proposed by Schiaffino et al. [80] provides personalized assistance to e-learning students. eTeacher observes a student's behavior and automatically builds the student's profile. This profile is comprised of the student's learning style and information about the student's performance for a given course, such as exercises done, topics studied, and exam results. A student's learning style is automatically detected from the student's actions in an e-learning system using Bayesian networks. eTeacher uses the information contained in the student profile to proactively assist the student by suggesting personalized courses of action that will help him or her during the learning process.

In the approach proposed by Lu et al. [60], learning material is recommended to users based on certain criteria like learning style, web browsing patterns, and other criteria, such as if the student is part-time or full-time, are taken into consideration. Users are judged based on the level of their knowledge. A learning material tree is built which is categorized into different levels and material is recommended according to the level of the student. This work does not provide search mechanism and there is no way where user could look for a material to study. Some of the notable work in

the area of recommender systems based on the user preferences was done by Rashid et al. [75]. They proposed a sequence of items for the collaborative filtering system to present to each new user for rating. They made use of information theory to select the items that will give the most value to the recommender system, aggregate statistics to select the items the user is most likely to have an opinion about and personalized techniques that predict which items a user will have an opinion about.

In the work proposed by Eyharabide et al. [43], the objective is to improve e-learning environment personalization, making use of users' preferences (e.g., the learning style of the user). They propose the AdaptWeb system, in which content and navigation recommendations are provided depending on the student's context. An e-learning environment for each user is personalized based on the information stored in a user profile.

6.2.2 Personalization Based on Ontology

Ontology is the relation defined between various concepts, some work done in building a course ontology was presented by Wali et al. [89], Chun et al.[32], Wali et al. [90] and SLOB [33]. Domain information about different courses like ontology, could also be used to derive personalized content for the user. Courseware Watchdog proposed by Tane et al. [82] allows making the most of the e-learning resources available on the Web. The tool addresses the different needs of tutors and learners and organizes their learning material according to their needs. Users can browse through web content, and the crawler finds the website and documents that match their interests. However, in this work, user preferences and knowledge are not taken into consideration.

Another work based on ontology by Markellou et al. [62] also takes personalization into consideration. The structure of knowledge and information plays a crucial role. The ontology-based organization helps managing of content related to a given course or lesson. The framework for personalization is based on usage profiles

of the users and the domain ontology. User information such as log files are used to record users' browsing activities. After this association, rules are calculated that have a support greater than a specified minimum support and confidence greater than a specified minimum confidence. Then the content from ontology is combined with users' navigation path.

Henze et al. [49], proposed a framework for personalized e-Learning in the semantic web and they show how the semantic web resource description formats can be utilized for automatic generation of hypertext structures from distributed metadata. Ontologies and metadata for three types of resources namely domain, user, and observation are investigated. User profile is built based on personal information.

6.2.3 Personalization in LMS and MOOCs

Despite being the most popular learning systems, LMSs provide limited support for personalization. LMSs, such as Intelligent Web Teacher [26], focuses on the concept of personalized e-Learning for the computer science (or informatics) education. They used Semantic Web technologies (e.g. ontologies) as a technological basis for personalization in e-learning. They proposed the Intelligent Web Teacher (IWT) which records user learning preferences and use ontology to model concepts that could be suggested according to user preferences and the evaluation received on each domain.

Alfanet [79], integrates the concepts of student modeling and personalization, but is not yet widely used. On the other hand, one of the most popular and frequently used Learning Management Systems, Moodle, offers limited support for personalization. It is possible to personalize the interface environment by creating new themes. In other words, specific activities can be made available to the learner according to certain conditions, such as the grade obtained in one or more tests, the completion of one or more activities, or a combination of the two. Teachers, however,

are responsible for defining possible alternative learning paths. Some MOOC systems provide recommendations on courses based on user interest.

6.3 Learning Data Model

User preference profile is built using learning abilities of a user or tracking browsing pattern for the user. However, little attention is paid to users' knowledge that acquired during the study. There is a need for a structure that defines precedence between the concepts to prepare a student for a given topic.

The learning model represents the data in graphical form, which helps to retain any precedence information. For personalizing the search responses, following are used:

Chapter precedence graph: The chapter precedence graph is used to derive a precedence relationship for the concepts covered in each of the chapters of the course textbook. In general, the textbook chapters are ordered and sometimes, the authors provide a guideline for presenting the topics to the students. This information can be used to build the "Chapter Precedence Graph", where each node corresponds to a chapter in the graph. The outgoing edges determine the child node or next chapters.

Course precedence graph: The course precedence graph models the prerequisite relationship between courses offered at a given institution.

User concept knowledge: The user concept knowledge represents the concepts that a student has covered from the courses she has taken. User concept knowledge is different for each user.

6.3.1 Chapter Precedence Graph

It is often essential that user understands the prerequisite concepts that provide background for the concept under study for a thorough understanding. This information is not easy to obtain as it requires expert knowledge. If a chapter C_1

precedes a chapter C_2, then it can be assumed that all the concepts covered in C_1 precede the concepts covered in C_2.

Every course has a prescribed textbook that provides an in-depth explanation of a course systematically. The course is divided into several chapters, where the initial chapters are usually introductory, and the later chapters are a comprehensive explanation of a specific topic. Chapter precedence graph can be built using the table of contents (TOC) of the textbook according to the structure of the textbook. In some textbooks, a chapter usage guideline is proposed to guide the instructors on possible orders to present the course topics (Figure 6.1). This guideline is useful in understanding the precedence level of chapters and in turn the concepts covered in each chapter.

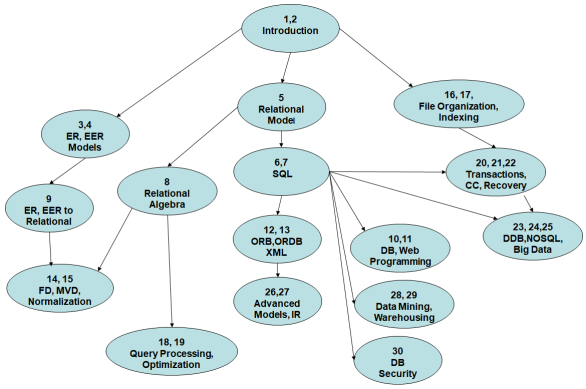


Figure 6.1 Chapter precedence graph, each vertex in the graph represents a chapter in the textbook. The precedence relation is represented by the edges between vertices.

Source: Fundamentals of Database Systems [42]

A chapter precedence graph $G_C = (V_C, E_C)$ is a graph where, the vertices V_C are the chapters (Chapter Titles) for a given course textbook and the edges E_C(Edges) represent the precedence relationship. Edges are added between the two vertices, if the vertices satisfy the precedence order. "$\prec$" is used to denote the precedence relationship. If a chapter $(V_i \prec V_j)$, then an edge is added from V_i to V_j.

Besides, each node (chapter) is associated with the concepts presented in that chapter in the graph. The index of the textbook can be used to obtain the information

about the location of each concept. A concept can appear in several pages/chapters, and the frequency is used to determine a home chapter for the concept.

6.3.2 Course Precedence Graph

The knowledge of a student regarding topics / concepts covered varies from one student to another. The course precedence graph depicts the relation between the courses available at a given institution. The course precedence graph represents the prerequisite relationship between courses. Although this is not always true, it can be assumed that a student masters all the concepts in the courses s/he has taken. The course precedence is defined as a graph $G_D = (V_D, E_D)$ where V_D(vertices) are the courses available within a University, E_D(Edges) are the directed edges that connects two vertices V_i and V_j if there exists a dependency between two courses. If course (V_D^i) is a prerequisite for course V_D^j $(V_D^i \prec V_D^j)$, then an edge is added from V_D^i to V_D^j as shown in figure 6.2.

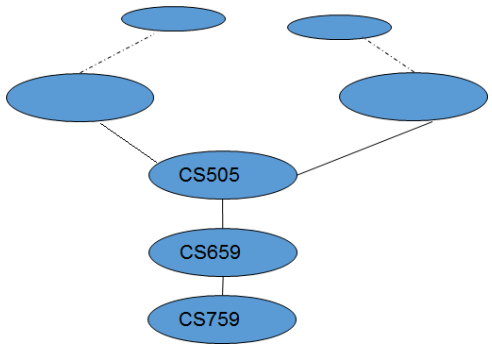


Figure 6.2 Example of course precedence graph. Each vertex in the graph is a course and the edges represent the prerequisite relation for each node.

In the example, the course CS759 has CS659 as a prerequisite. CS659 has CS505 as a prerequisite. If a user is registered for a course CS759, it can be assumed that the user must have taken or possessed knowledge of courses CS659 and CS505.

6.3.3 User Concept Knowledge

Each student takes several courses during study towards a degree. A separate concept index is created for every user to represent the concept information for every user. Personalized results consider the knowledge of the user. There are two different ways to represent user concept knowledge:

1. A list of course precedence graphs where each graph represents the course user has already taken.

2. A list of concepts, supported by an index to represent the concepts user has already covered.

The structural information for user concept knowledge is not necessary to determine if the user is aware of the concept. Hence, the user concept knowledge is represented as a list of concepts as follows:

$$\begin{aligned} C_u &= C_1, C_2 \dots C_n \\ K_u &= V_c^1, \dots V_c^n \end{aligned} \tag{6.1}$$

where, C_u is a list of courses user has taken. K_u is the user concept knowledge list, V_c^1 are the vertices (concepts) covered by user corresponding to the chapter precedence graph for course C_1

6.4 Indexing, Query Processing and Ranking

Chapter precedence graphs containing a query term are extracted with the help of an index. In the query processing step, the top k chapter precedence graphs are retrieved according to the scores. The concepts that needed to be studied are represented as the subgraphs extracted from the chapter precedence graph. Each subgraph consists of a set of nodes with the leaf node as the chapter covering the query term and all the

parent nodes connected to the leaf node. The matching step considers the knowledge of the user to display the personalized results.

6.4.1 Indexing

The index representation can be given as follows:

Information, such as textbook title, corresponding chapter precedence graph and

```
<Textbook title = 'textbookTitle' precedenceGraph='GraphId'>
   <Chapter chapternumber = 1, presentationTitle='Ptitle'>
      <ChapterTitle> title </ChapterTitle>
      <SectionTitle> title1, title2 </SectionTitle>
      <ChapterText> text</ChapterText>
      <PageNumber>1...Pn</PageNumber>
    </Chapter>
</Textbook>
```

Figure 6.3 Textbook representation

chapters are recorded for each textbook. For each chapter, information, such as chapter title, chapter text, section titles and associated page numbers are recorded. This information (Figure 6.3), is then given as input to Apache Lucene to build inverted indexes.

6.4.2 Query Results

Top-k chapter precedence graphs containing the concept are retrieved (6.3) as a result of the user query. For each chapter precedence graph, chapter representing each term is also recorded. All the chapters preceding the current chapter in the chapter precedence graph form the induced subgraph, and the rest of the chapters are disregarded. If a user queries for the term "big data" (Figure 6.4), which appears in chapter 25, the induced subgraph will contain three paths, $\{chapter\{1,2\},\{16,17\},\{20,21,22\},\{23,24,25\}\}$, $\{chapter\{1,2\},\{5\},\{6,7\},\{23,24,25\}\}$, $\{chapter\{1,2\},\{5\},\{6,7\},\{20,21,22\},\{23,24,25\}\}$.

There are two different ways to represent query graph at this point

1. List of sub-graphs.

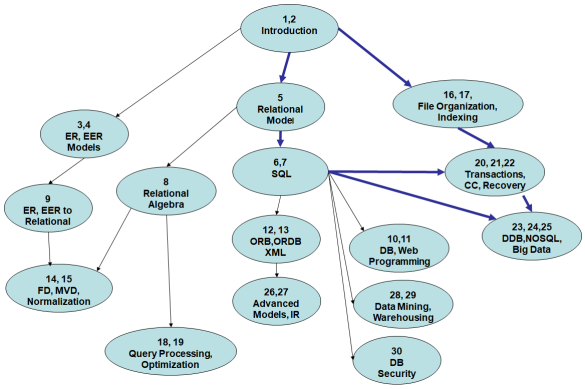


Figure 6.4 Induced subgraph for user query "big data."

2. Merge the graphs in the query to form a single graph as a query graph.

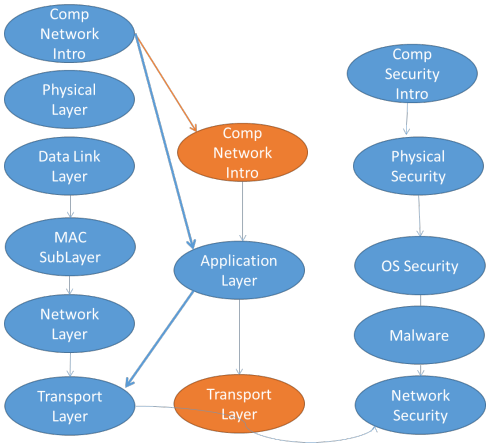


Figure 6.5 Query graph merged to single graph.

In the figure(6.5), three Course precedence graphs are returned as a result of query "TCP". Here similar concepts, such as computer network info and transport layer can be merged with the left subgraph. And since the query term resulted in the chapters at the leaf level in the figure, a link can be added between them too.

Since merging of graphs is a research issue, and different courses do not align the concepts in similar order, the merging operation could make it expensive and unnecessary.

Instead, the query using approach (1) is used to retrieve only top k subgraphs corresponding to k separate sub-queries; this keeps the retrieval constant and less expensive.

6.4.3 Course Material Retrieval and Ranking

In general, a user query $q =< t_1, t_2....t_m >$ consists of one or more terms. Each course chapter is given a score based on the equation below. Given a query (q) which is list of keywords. The score of a textbook chapter for a given term is determined following the classical TF/IDF formulas as follows:

$$\begin{aligned} Score(q, ch) = &\sum_{t \in q} tf(t \ in \ ch) \times idf(t) \times weight_{chaptertitle} \\ &+ \sum_{t \in q} tf(t \ in \ ch) \times idf(t) \times weight_{sectiontitle} \\ &+ \sum_{t \in q} tf(t \ in \ ch) \times idf(t) \times weight_{body} \end{aligned} \tag{6.2}$$

where, q is the query consisting of one or more terms,ch, is the chapter of textbook that contains the term, $weight_{chaptertitle}$, $weight_{sectiontitle}$ *and* $weight_{body}$ are the weights added if keyword appears in the chapter title, section title and body of the textbook respectively, $t_f(t \in ch)$ is the term frequency of the term within the chapter, *idf(t)* is the inverse document frequency given by $\log \frac{N}{d_f}$.

Each chapter is a considered a document. A cumulative score of the textbook is computed as an aggregate score of individual chapters covering the same topic within a textbook. The most representative chapter is also recorded for each keyword.

Top-k chapter precedence graphs containing the concept are retrieved (Equation 6.3) as a result of the user query. For each chapter precedence graph, chapter representing each term is also recorded.

$$\Psi(q) = \{C_i^t\} \tag{6.3}$$

where, $\Psi(q)$ is the mapping function that maps query term to list of textbooks (t) that contain the term and most relevant chapter(i). These textbooks are ordered by score (S_C). The list of chapter precedence graphs can be obtained from the index (Equation 6.3). Function "ϕ" is defined to extract the subgraph from each chapter precedence graph. Chapter representing each term is identified for each chapter precedence graph. All the chapters preceding the current chapter in the chapter precedence graph form the induced subgraph, rest of the chapters are discarded.

$$\phi(G_C, C) = \{G'_C : G'_C \subseteq G_C\} \tag{6.4}$$

where, $\phi(G^i_C, C^i)$ is the function that extracts the subgraph from the graph for a given term. $G'_C = (V'_C, E'_C) : V'_C \subseteq V_C, E'_C \subseteq E_C$.

List of subgraphs is used to represent the query (L_q) for the matching step.

$$L_q = \{G^{i'}_C, ...G^{k'}_C\} \tag{6.5}$$

where, k is the number representing k^{th} graph of top k graphs, $G^{1'}_C....G^{k'}_C$ are the k subgraphs (subset of corresponding course precedence graphs) returned by the query.

6.5 Matching Query Resultant Graph and User Graph

The matching step considers the knowledge of the user to display the personalized results. The results obtained from the query (L_q) are matched against the user concept knowledge (K_u).

The user graph K_u consists of all the courses the user has covered and can be represented as:

$$K_u = \{G^i_C, G^j_C....G^n_C\} \tag{6.6}$$

where, G^i_C is the chapter precedence graph of the particular course i. The concepts the user needs to cover is the result obtained from (Equation 6.5).

A matching function ("λ") represents the similar concepts between the query results and the user concept knowledge as follows:

$$\lambda(L_q, K_u) : L_q \cap K_u \tag{6.7}$$

where the vertices in V_q and V_u match if they have same title.
The concepts left to be covered are given as follows:

$$\{L_q \setminus K_u\} \tag{6.8}$$

where, the distinct vertices of L_q and K_u are included in the set.
Since the representation of concepts is different in various courses, the concepts are presented to the user in the color-coded form. The *blue* color indicates, the concepts that are already covered by the user, and the *red* color indicates the concepts left to be covered by the user. The recommendations of the concepts to be covered, are presented to the user. The users can browse through the recommended concepts to prepare thoroughly. The recommended concepts are linked to the lecture material that covers them, such as slide, videos and textbook.

6.6 Examples of Queries

The data set consisted of four different courses, each course has a chapter precedence graph.
$G_C = \{G_C^1, G_C^2, G_C^3, G_C^4\}$
Query1 (q1):"tcp"
The UCS application (based on G_C^2) returns the slides that cover the keyword "tcp". Along with slides, videos and textbook chapter that cover the concept are also listed.
Out of four graphs in the data set, three contain the concept "tcp" (as shown in the Figure 6.7). Result Intermediate:$\{G_C^1, G_C^2, G_C^3\}$
Next, we pick top k subgraphs from this list (k=2) in this case.

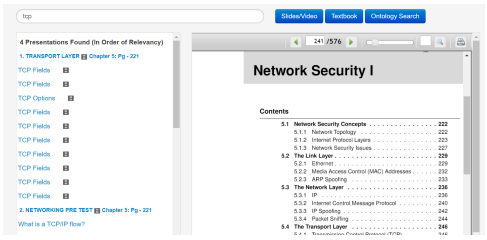


Figure 6.6 Results of query "Transmission Control Protocol (tcp)," on the UCS system.

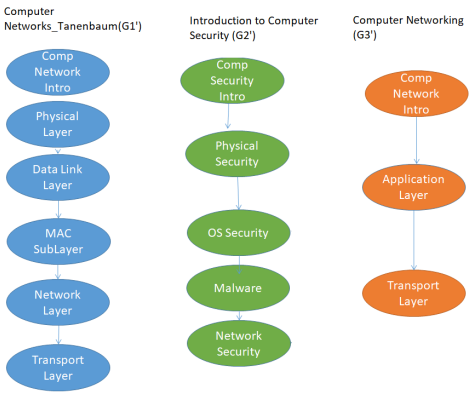


Figure 6.7 Induced subgraph for query "tcp".

Result top k: $\{G_C^1, G_C^2\}$

The subgraph for each of them consists of precedence graph from the root to the chapter that contains this term. This becomes the leaf node.

Subgraphs extracted: $\{G_C^{1'}, G_C^{2'}\}$ The query results contain prerequisite concepts for tcp.

Concepts needed to be covered:

G_q = {computer network introduction, physical layer, data link layer, medium access control sublayer, network layer,

transport lay, introduction computer security introduction, physical security, operating systems security, malware, network security, network security ii}. Following cases describe personalized results under different scenarios, based on user experience. The concepts are recommended in a color-coded manner. The *blue* color indicates, the

concepts that are already covered by the user, and the *red* color indicates the concepts left to be covered by the user.

- Case 1: User does not have any relevant experience to the topic in query

  Query "network security"

  Query result: {computer network introduction, physical layer, data link layer, medium access control sublayer, network layer, transport layer, application layer, network security}

  User 1 course list: Database

  Concepts already covered: {none}

  Suggested concepts to be covered: {computer network introduction, physical layer, data link layer, medium access control sublayer, network layer, transport layer, application layer, network security}

  In this case, the user has no experience related to query in question. The user is registered for a course "Database", and has no prior experience related to network security. The precedence subgraph G1' is suggested as the result of the query.

- Case 2: User has some relevant experience to the topic in query

  Query: "security" Query result: G2', G1', G4'

  introduction computer security introduction, physical security , operating systems security , computer network introduction, physical layer, data link layer, medium access control sublayer, network layer, transport layer, application layer, network security, databases, database system concepts architecture , relational data model, basic sql , more sql: complex queries triggers views schema modification, database security

  User 2 course list: Database

  Concepts already covered:{ databases, database system concepts architecture , relational data model, basic sql , more sql: complex queries triggers views

schema modification, database security}

Suggested concepts to be covered: {introduction computer security introduction, physical security, operating systems security, computer network introduction, physical layer, data link layer, medium access control sublayer, network layer, transport layer, application layer, network security }

In case 2, the query "security" is a generic term. The security could be related to network, database, etc. Top k (top 3) results fetched from the database (G2',G1',G4'). Since the user has already covered "Database" course, the chapter precedence subgraph (G4') is excluded from the result and G2' and G1' are recommended to the user.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

In this dissertation, we presented "Ultimate Course Search", an interface that provides a simple, easy-to-use interface, but also a highly beneficial way to search various lecture material. For enabling search on the learning material, we index the three most widely used lecture media, namely slides, videos and textbooks. We index the slides by identifying relevant keywords from the slide.

We show that without manually annotating the slides and videos, we can effectively link the material by indexing the video segments with the corresponding slide. We present a comparative analysis of the state-of-the-art techniques to determine the feature descriptors most suitable for detecting transitions of learning video and conclude that HOG descriptors performed the best. Our results show that detecting the transitions automatically along with matching the original slides helps us to identify transitions accurately.

We presented a localization technique to extract the slide from the video frame in the case of classroom videos that capture the surroundings as well. Our results show that the transition detection performance was improved by using localization. We also compare the results to saliency technique and show that our localization technique performs better for educational videos.

UCS also offers search in the textbook by indexing the content from the back-index of the textbook along with the page number information. UCS thus integrates these three learning media into a single platform, which provides students with a way to learn the material effectively and efficiently. The results presented to users after a keyword search are based on the region where the keyword appears, displaying the results in such a fashion brings the most important and relevant content to the top.

Finally, we proposed a technique to personalize the content for users based on knowledge of an individual user. We show that by representing the concepts as a precedence graph, a user is presented with concepts that are needed to be covered to understand a concept fully. We also take user knowledge into account by representing user knowledge in the form of a user graph.

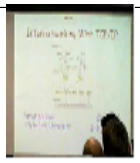
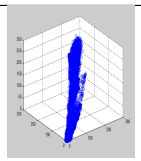
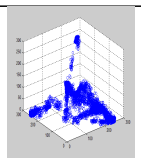
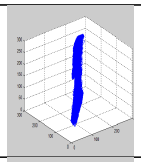
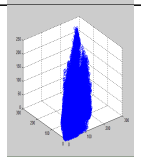
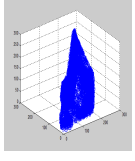
In future work, the speech of an instructor can be analyzed, since the speech includes plenty of keywords that are otherwise not present in the presentation slides. Audio obtained from the video can be converted to text using the speech-to-text software. Adding the keywords from the speech of the instructor would make the interface for videos independent of slide information. Additional video interfaces could be added to UCS for searching the videos with keywords obtained from speech.

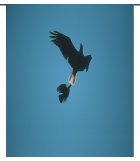
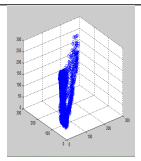
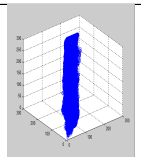
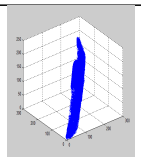
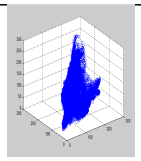
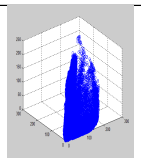
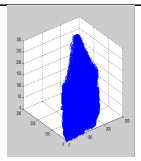
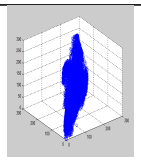
Currently, user data was created for the experiments and not integrated into the UCS system. In the future, actual user data can be used for personalized learning. The content can be personalized based on learning preference of the user using the UCS application. Chapter precedence graph was used to create the precedence relation between the concepts at the chapter level. Slides can be mapped to the textbook and used to develop refined levels of precedence between concepts.

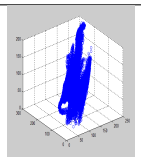
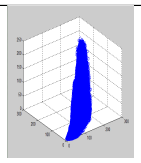
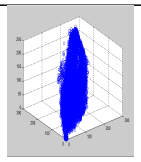
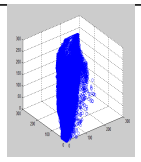
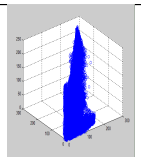
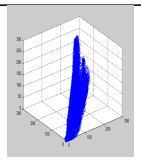
APPENDIX A

SIMILARITY MEASURES FOR DCT, MARGINAL AND GRAYSCALE

Table A.1: Similarity Measures (Jaccard Index (JI), F-measure (FM)) computed for DCT, Marginal, and Grayscale space

Image Id	*Original Image*	*3D Distribution*	*DCT*	*Marginal*	*Grayscale*
(1*)			JI: 0.09 FM: 0.57 Time: 2.04	JI: 0.09 FM: 0.72 Time: 1.74 Somedian = 16	JI: 0.01 FM: 0.20 Time: 0.98
(2*)			JI: 0.66 FM: 0.96 Time: 0.77	JI: 0.54 FM: 0.95 Time: 0.61 S0median : 241	JI: 0.26 FM: 0.86 Time: 0.32
(253036)			JI: 0.13 FM: 0.92 Time: 2.81	JI: 0.08 FM: 0.44 Time: 1.84	JI: 0.07 FM: 0.34 Time: 1.96
(118020)			JI=0.05, FM = 0.49, Time: 6.06	JI=0.06, FM = 0.59 Time: 4.89 S0median : 27 Ideal: 101	JI=0.05, FM = 0.52, Time: 5.55
(118035)			JI=0.13, FM = 0.74 Time: 3.536	JI=0.12, FM = 0.85 Time: 2.766 S0median :43 Ideal: >151	JI=0.23, FM = 0.90 Time: 2.038

(135069)			JI=0.39, FM = 0.96 Time: 2.649	JI=0.55, FM = 0.68 Time: 1.8617 S0median : 81 Ideal : 51 - 151	JI=0.51, FM = 0.84 Time: 1.824
(138078)			JI=0.09, FM - 0.70 Time: 4.59	JI=0.09, FM = 0.86 Time: 2.321 S0median 16 Ideal: 51	JI=0.06, FM = 0.56 Time: 3.6011
(161062)			JI=0.23, FM = 0.97 Time: 4.22	JI=0.10 FM = 0.23 Time: 1.436 S0median 30 Ideal: 1 - 100	JI=0.12, FM = 0.96 Time: 3.70
(172032)			JI: 0.06 FM: 0.71 Time: 4.87	JI: 0.14 FM: 0.89 Time: 2.71 Somedian : 29 Ideal:1 –50	JI: 0.09 FM: 0.80 Time: 3.22
(187029)			JI: 0.09 FM: 0.77 Time: 4.87	JI: 0.12 FM: 0.94 Time: 2.06 Somedian = 33	JI: 0.11 FM: 0.87 Time: 2.32
(189003)			JI: 0.01 Time: 5.2	JI: 0.07, FM: 0.69 Time: 3.16 S0median 27, Ideal: 1-50	JI : 0.08, FM : 0.58 Time: 3.8
(24004)			JI=0.05, FM = 0.42 Time: 5.7	JI=0.05, FM = 0.47 Time: 3.73 S0median 37	JI=0.037, FM = 0.43 Time: 4.6

(299091)			JI: 0.11 FM: 0.65 Time: 3.52	JI: 0.13 FM: 0.79 Time: 1.81 Somedian : 43 Ideal: 1-100	JI: 0.13 FM: 0.79 Time: 1.81 Somedian : 43 Ideal: 1-100
(302003)			JI: 0.22 FM: 0.94 Time: 3.6	JI: 0.26 FM: 0.97 Time: 2.48 Somedian : 46 Ideal:101	JI: 0.34 FM: 0.81 Time: 2.33
(323016)			JI: 0.06 FM: 0.49 Time: 3.8	JI: 0.13 FM: 0.68 Time: 3.26 Somedian :26 Ideal: 1 – 50	JI: 0.09 FM:0.62 Time: 2.5
(368078)			JI=0.08, FM = 0.71 Time: 5.7	JI=0.08, FM = 0.64 Time: 4.202 S0median 60, Ideal: 50 – 100	JI=0.10 FM = 0.76 Time: 3.86
(372047)			JI=0.03, FM = 0.42 Time: 4.754	JI=0.04 FM = 0.494107 Time: 3.7301 S0median : 44 Ideal: 1-50	JI=0.05, FM = 0.54 Time: 3.293
(372047)			JI=0.22, FM = 0.94 Time: 6.3512	JI=0.19, FM = 0.69 Time: 3.086 S0median 32 Ideal: 101	JI= 0.22 , FM = 0.88, Time: 3.75

BIBLIOGRAPHY

[1] Blackboard lms. http://www.blackboard.com/learning-management-system/blackboard-learn.aspx. Accessed Sept 30, 2017.

[2] Comparing xmoocs and cmoocs philosophy and practice. http://www.tonybates.ca/2014/10/13/comparing-xmoocs-and-cmoocs-philosophy-and-practice/. Accessed Sept 30, 2017.

[3] Comparing xmoocs and cmoocs: philosophy and practice. http://www.slideshare.net/josias20/massive-open-online-courses-moocs. Accessed Sept 30, 2017.

[4] From maths class on yahoo doodle to a free world-class education for everyone – khan academy. http://www.fedena.com/blog/2013/09/from\-maths\-class\-on\-yahoo\-doodle\-to\-a\-free\-world\-class\-education\-for\-everyone\-khan\-academy.html. Accessed Sept 30, 2017.

[5] itunes u. https://itunes.apple.com/us/app/itunes-u/id490217893?mt=8. Accessed Sept 30, 2017.

[6] Moodle. https://moodle.org/. Accessed Sept 30, 2017.

[7] Open university. http://www.open.ac.uk/. Accessed Sept 30, 2017.

[8] Peer 2 peer university(p2pu). https://www.p2pu.org/en/. Accessed Sept 30, 2017.

[9] Udemy. http://www.pcmag.com/article2/0,2817,2483851,00.asp. Accessed Sept 30, 2017.

[10] University of catalonia. http://www.upc.edu/?set_language=en. Accessed Sept 30, 2017.

[11] edx. https://www.edx.org/, 2012. Accessed Sept 30, 2017.

[12] G. D. Abowd. Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal*, 38(4):508–530, Dec 1999.

[13] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 1597–1604, Jun 2009.

[14] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, May 2012.

[15] J. Adcock, M. Cooper, L. Denoue, H. Pirsiavash, and L. A. Rowe. Talkminer: a lecture webcast search engine. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 241–250, Oct 2010.

[16] J. Adcock, A. Gigensohn, M. Cooper, T. Liu, L. Wilcox, and E. Rieffel. Fxpal experiments for trecvid 2004. In *Text Retrieval Conference Video Retrieval 2004 Workshop*, Nov 2004.

[17] A. Amir, W. Hsu, G. Iyengar, C.Y. Lin, M. Naphade, A. Natsev, C. Neti, H.J. Nock, J.R. Smith, B.L. Tseng, Y. Wu, and D. Zhang. Ibm research trecvid 2003 video retrieval system. In *Text Retrieval Conference Video Retrieval 2003 Workshop*, Nov 2003.

[18] Apache. Apache poi - the java api for microsoft documents. `http://poi.apache.org/`. Accessed Sept 30, 2017.

[19] F. Arman, A. Hsu, and M. Chiu. Image processing on encoded video sequences. *Multimedia Systems*, 1(5):211–219, Mar 1994.

[20] J. Baber, N. Afzulpurkar, M. N. Dailey, and M. Bakhtyar. Shot boundary detection from videos using entropy and local descriptor. In *17th International Conference on Digital Signal Processing (DSP)*, pages 1–6, Jul 2011.

[21] D. Bargeron, J. Grudin, A. Gupta, and E. Sanocki. Annotations for streaming video on the web: system design and usage studies. In *Proceedings of the Eight International World Wide Web Conference*, pages 61–75, Mar 1999.

[22] T. Bay, H.and Tuytelaars and L. Van Gool. *SURF: Speeded Up Robust Features*, volume 3951, pages 404–417. May 2006.

[23] P. Brusilovsky. Web-based education for all: A tool for development adaptive courseware. *Computer Networks*, 30(1-7):291–300, Apr 1998.

[24] L. Busin, J. Shi, N. Vandenbroucke, and L. Macaire. Color space selection for color image segmentation by spectral clustering. In *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 262–267, Nov 2009.

[25] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8 (6):679–698, Nov 1986.

[26] N. Capuano, M. Gaeta, A. Micarelli, and E. Sangineto. An intelligent web teacher system for learning personalization and semantic web compatibility. In *11th International PEG Conference Powerful ICT for Teaching and Learning*, Jun 2003.

[27] T. Carron. Segmentation d'images couleur dans la base teinte-luminance-saturation: approche numehrique et symbolique, 1995.

[28] Z. Cernekova, C. Kotropoulos, and I. Pitas. Video shot segmentation using singular value decomposition. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, volume 2, pages 301–302. IEEE, Apr 2003.

[29] C. Chen. Intelligent web-based learning system with personalized learning path guidance. *Computers & Education*, 51(2):787 – 814, 2008.

[30] C. Chen, H. Lee, and Y. Chen. Personalized e-learning system using item response theory. *Computers & Education*, 44(3):237 – 255, Apr 2005.

[31] S.K. Choubey and V.V. Raghavan. Generic and fully automatic content-based image retrieval using color. *Journal of Pattern Recognition Letters*, 18 (11-13):1233–1240, Nov 1997.

[32] S. Chun and J. Geller. Developing a pedagogical cybersecurity ontology. pages 117–135, Jan 2015.

[33] S. A. Chun, J. Geller, A. Taunk, K. Sankaran, and T. Swaminathan. Slob: Security learning by ontology browsing: Comprehensive cyber security learning resources in a web portal. *Journal of Computing Sciences in Colleges*, 31(5):95–101, May 2016.

[34] E. Cooke, P. Ferguson, G. Gaughan, C. Gurrin, G. Jones, H. L. Borgue, H. Lee, S. Marlow, K. McDonald, M. McHugh, N. Murphy, N. O'Connor, N. O'Hare, S. Rothwell, A. Smeaton, and P. Wilkins. Trecvid 2004 experiments in dublin city university. In *Text Retrieval Conference Video Retrieval 2004 workshop*, Nov 2004.

[35] Coursera. `https://www.coursera.org/`, 2011. Accessed Sept 30, 2017.

[36] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, Jun 2005.

[37] P. Dolog, N. Henze, W. Nejdl, and M. Sintek. The personal reader: Personalizing and enriching learning resources using semantic web technologies. In *Third International Conference Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 3137, pages 85–94, Aug 2004.

[38] P. Dolog, N. Henze, W. Nejdl, and M. Sintek. Personalization in distributed e-learning environments. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 170–179, May 2004.

[39] C. Dorai, V. Oria, and V. Neelavalli. Structuralizing educational videos based on presentation content. In *Image Processing, 2003, ICIP 2003*, volume 3, pages 1029–1032. IEEE, Sept 2003.

[40] Stephen Downes. Connectivism and connective knowledge. `http://www.downes.ca/post/58207`. Accessed Sept 30, 2017.

[41] R. Dugad, K. Ratakonda, and N. Ahuja. Robust video shot change detection. In *IEEE Second Workshop on Multimedia Signal Processing*, pages 376–381, Dec 1998.

[42] R. Elmasri and S. Navathe. *Fundamentals of Database Systems*. Pearson Education Limited, 2010.

[43] I. Eyharabide, V.and Gasparini, S. Schiaffino, M. Pimenta, and A. Amandi. *Personalized e-Learning Environments: Considering Students' Contexts*, pages 48–57. Springer Berlin Heidelberg, 2009.

[44] C. Foley, C. Gurrin, G. Jones, H. Lee, S. McGivney, N. E. O'Connor, S. Sav, A. F. Smeaton, and P. Wilkins. Trecvid 2005 experiments in dublin city university. In *Text Retrieval Conference Video Retrieval 2005 workshop*, Nov 2005.

[45] W. Gerhard and M. Specht. User modeling and adaptive navigation support in www-based tutoring systems. In *User Modeling: Proceedings of the Sixth International Conference UM97 Chia Laguna*, pages 289–300, Jun 1997.

[46] M. Goodrich and R. Tamassia. *Introduction to Computer Security*. Pearson Education Limited, Harlow, England, 2014.

[47] S.H. Han, K.J Yoon, and I.S. Kweon. A new technique for shot detection and key frames selection in histogram space. In *Workshop on Image Processing and Image Understanding*, Apr 2000.

[48] A.G. Hauptmann, R. Baron, M.Y Chen, M. Christel, P. Duygulu, C. Huamg, R. Jin, W.H. Lin, T. Ng, N. Moraveji, N. Papernick, C. Snoek, G. Tzanetakis, J. Yamg, R. Yan, and H. Wactlar. Informedia at trecvid 2003: Analyzing and searching broadcast news and video. In *Text Retrieval Conference Video Retrieval 2003 workshop*, Nov 2003.

[49] N. Henze, P. Dolog, and W. Nejdl. Reasoning and ontologies for personalized e-learning in the semantic web. *Educational Technology & Society*, 7(4):82–97, Oct 2004.

[50] J. Hunter and S. Little. Building and indexing a distributed multimedia presentation archive using smil. In *Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries*, pages 415–428, Sept 2001.

[51] J. Jin and R. Wang. The development of an online video browsing system. In *Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing*, volume 11, pages 3–9, May 2001.

[52] V.K. Kamabathula and S. Iyer. Automated tagging to enable fine-grained browsing of lecture videos. In *IEEE International Conference on Technology of Education*, pages 96–102, Jul 2011.

[53] T. Kikukawa and S. Kawafuchi. Development of an automatic summary editing system for the audio visual resources. In *Transactions of the Institute of Electronics, Information and Communication Engineers*, volume 75 (2), pages 204–212, 1992.

[54] J. Li, Y. Ding, W. Li, and Y. Shi. Dwt- based shot boundary detection using support vector machine. volume 1, pages 214–221, Aug 2009.

[55] J. Li, Y. Ding, Y. Shi, and W. Li. A divide-and-rule scheme for shot boundary detection based on sift. *International Journal of Digital Content Technology and its Applications*, 4(3):202–214, Jun 2010.

[56] R. J. Lienhart. Comparison of automatic shot boundary detection algorithms. In *In Proceedings of Storage and Retrieval for Image and Video Databases, SPIE*, volume 3656, pages 290–301, Dec 1998.

[57] M. Liška, V. Rusňák, and E. Hladká. Automated hypermedia authoring for individualized learning, Sept 2007.

[58] T. D. C. Little, G. Ahanger, R. J. Folz, J. F. Gibbon, F. W. Reeve, D. H. Schelleng, and D. Venkatesh. A digital on-demand video service supporting content-based queries. In *Proceedings of the First ACM International Conference on Multimedia*, pages 427–436, Aug 1993.

[59] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[60] J. Lu. A personalized e-learning material recommender system. pages 23–28, Jan 2004.

[61] Y. Ma and H. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of the Eleventh ACM International Conference on Multimedia*, pages 374–381, Nov 2003.

[62] P. Markellou, I. Mousourouli, S. Spiros, and A. Tsakalidis. Using semantic web mining technologies for personalized e-learning experiences. In *Proceedings of the Web-based Education*, pages 461–826, Feb 2005.

[63] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings 8th International Conference in Computer Vision*, volume 2, pages 416–423, Jul 2001.

[64] MIT. Mit open courseware. `http://ocw.mit.edu/index.htm`, 2001. Accessed Sept 30, 2017.

[65] moocnewsandreviews. A short history of moocs and distance learning. http://moocnewsandreviews.com/a-short-history-of-moocs-and-distance-learning/. Accessed Sept 30, 2017.

[66] moocnewsandreviews. What is a massive open online course anyway? http://moocnewsandreviews.com/what-is-a-massive-open-online-course-anyway-attempting-definition/. Accessed Sept 30, 2017.

[67] S. Mukhopadhyay and B. Smith. Passive capture and structuring of lectures. In *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)*, pages 477–487, Oct 1999.

[68] A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. In *Proceedings of the IFIP TC2/WG 2.6 Second Working Conference on Visual Database Systems II*, pages 113–127, Sept 1992.

[69] C.W. Ngo, T.C. Pong, and R.T. Chin. Video partitioning by temporal slice coherency. In *Circuits and System for Video Technology*, volume 11, pages 941–953. IEEE, Aug 2001.

[70] University of Illinois. Programmed logic for automatic teaching operations (plato). https://en.wikipedia.org/wiki/PLATO_(computer_system). Accessed Sept 30, 2017.

[71] K. Otsuji, Y. Tonomura, and Y. Ohba. Video browsing using brightness data. In *Proceedings SPIE 1606: Visual Communications and Image Processing*, volume 1606, pages 980–989, Nov 1991.

[72] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *Proceedings of the Fourth ACM International Conference on Multimedia*, pages 65–73, Nov 1996.

[73] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 733–740, Jun 2012.

[74] S. Rajgure, V. Oria, K. Raghavan, H. Dasadia, S. S. Devannagari, R. Curtmola, J. Geller, P. Gouton, E. Renfro-Michel, and S. A. Chun. Ucs: Ultimate course search. In *14th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–3, Jun 2016.

[75] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. M. McNee, J. A. Konstan, and J. Riedl. Getting to know you: Learning new user preferences in recommender systems. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, pages 127–134, Jan 2002.

[76] S. Repp, A. GroB, and C. Meinel. Browsing within lecture videos based on the chain index of speech transcription. 1(3):145–156, Dec 2008.

[77] L. A. Rowe and J. M. Gonzlez. Bmrc lecture browser demo, 1999.

[78] L.A. Rowe, D. Harley, P. Pletcher, and S. Lawrence. Bibs: A lecture webcasting system, Mar 2001.

[79] O. C. Santos, E. Gaudioso, C. Barrera, and J. Boticario. Alfanet: An adaptive e-learning platform. In *2nd International Conference on Multimedia and ICTs in Education (m-ICTE2003)*, Dec 2003.

[80] S. Schiaffino, P. Garcia, and A. Amandi. eteacher: Providing personalized assistance to e-learning students. *Computers & Education*, 51(4):1744 – 1754, Dec 2008.

[81] B. Shahraray. Scene change detection and content-based sampling of video sequences. In *Proceedings SPIE Digital Video Compression: Algorithms and Technologies*, volume 2419, pages 2–13, Apr 1995.

[82] J. Tane, C. Schmitz, and G. Stumme. Semantic resource management for the web: An e-learning application. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*, pages 1–10, May 2004.

[83] Techsmith. Camtasia studio. `http://www.techsmith.com/camtasia.html`. Accessed Sept 30, 2017.

[84] S. Thrun, D. Stavens, and M. Sokolsky. Udacity. `https://www.udacity.com/`, 2012. Accessed Sept 30, 2017.

[85] A. Totterdell. An algorithm for detecting and classifying scene breaks in mpeg video bit streams, 1998.

[86] W. Tsai. Moment-preserving thresolding: A new approach. *Computer Vision, Graphics, and Image Processing*, 29(3):377 – 393, 1985.

[87] M. Turoff. Telecommunications: Meeting through your computer: Information exchange and engineering decision-making are made easy through computer-assisted conferencing. *IEEE Spectrum*, 14(5):58–64, May 1977.

[88] H. Ueda, T. Miyatake, and S. Yoshizawa. Impact: An interactive natural-motion-picture dedicated multimedia authoring system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 343–350, Apr 1991.

[89] A. Wali, S. A. Chun, and J. Geller. A bootstrapping approach for developing a cyber-security ontology using textbook index terms. In *2013 International Conference on Availability, Reliability and Security*, pages 569–576, Sept 2013.

[90] A. Wali, S. A. Chun, and J. Geller. A hybrid approach to developing a cyber security ontology. In *Proceedings of DATA 2014, 3rd International Conference for Data Management Technologies and Applications*, pages 377–384, Aug 2014.

[91] H. Yang, C. Oehlke, and C. Meinel. A solution for german speech recognition for analysis and processing of lecture videos. In *2011 10th IEEE/ACIS International Conference on Computer and Information Science(ICIS)*, pages 201–206, May 2011.

[92] H. Yang, M. Siebert, P. Luhne, H. Sack, and C. Meinel. Lecture video indexing and analysis using video ocr technology. In *2011 Seventh International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*, pages 54–61, Nov 2011.

[93] J. Yu and M. D. Srinath. An efficient method for scene cut detection. *Pattern Recognition Letters*, 22(13):1379–1391, Nov 2001.

[94] R. Zabih, J. Miller, and K. Mai. A feature-based algorithm for detecting and classifying scene breaks. In *Proceedings of the Third ACM International Conference on Multimedia*, pages 189–200, Nov 1995.

[95] H. Zhang, A. Kankanhalli, and S.W. Smoliar. Automatic partitioning of full-motion video. *ACM Multimedia System*, 1(1):10–28, Jan 1993.