

Using Advanced Computing Techniques to Implement a Distance  
Education System

by

Michael N. Wallick

A thesis submitted in partial fulfillment of the requirements  
for Honors in the Major  
in the College of Engineering and Computer Science  
at the University of Central Florida  
Orlando, Florida

Spring Term  
2001

Major Professor:  
Niels da Vitoria Lobo

© 2001 by Michael N. Wallick

## Abstract

As more universities begin to offer distance education classes, advances in current methods of delivering classroom information must be introduced. At present, universities use two different methods for distance education. The first is text-based web pages, which due to bandwidth restrictions are generally unable to display complex multimedia information. The second method is to videotape lectures and distribute the tapes to distant sites. While this does a reasonable job of simulating a classroom, the cost associated with producing and distributing the videos and the delay involved in distribution makes this system unattractive.

This thesis presents a method for compressing the classroom video to a smaller size so that the lecture can be rebroadcast over the Internet without losing classroom information. In addition, methods will be demonstrated for automatically extracting various types of information from a videotaped lecture; this will result in a more interactive lecture than a simple videotape would provide.

## Acknowledgments

I would like to thank many people and organizations whom, without their help, this thesis would not be what it is.

The National Science Foundation (NSF) for funding the Research Experience for Undergraduates (REU) program and School of Electrical Engineering and Computer Science at the University of Central Florida for supporting the REU program, under which all of my research was conducted.

The FEEDS program at UCF, including the staff for providing me with video-taped lectures to use as data, as well as the students in the classes for their insightful and informative feedback of the results.

The Honor College at UCF, for creating the Honors in the Major program, as well as helping me with much of my research.

My entire family for the support that they have shown me throughout the entire project; despite the fact that they did not understand most of it.

All of my friends who have stood by me and listened to me talk about my thesis, especially Mike and Christine. Mike for putting up with me and the complaints that this project would generate for the past three years, and Christine for using her own spare time to help me record that “extra little bit” of data.

Last, and certainly not least, my thesis committee. Without their dedication, guidance and support I would not be able to produce this thesis. I would especially like to thank my committee chair, Dr. Niels da Vitoria Lobo. For the past 4 years, Dr. Lobo has been my advisor, my mentor, and my friend. I owe much

of my current and future success to Dr. Lobo. I will always remember everything that he has done for me.

## TABLE OF CONTENTS

List of Figures . . . . .	ix
<b>1 Introduction . . . . .</b>	<b>1</b>
1.1 Statement of Problem . . . . .	1
1.2 Audience . . . . .	2
1.3 Structure of the Thesis . . . . .	4
<b>2 Previous Work . . . . .</b>	<b>6</b>
<b>3 Extracting Key frames from a Videotaped Lecture . . . . .</b>	<b>8</b>
3.1 Extracting the Key frames . . . . .	9
3.2 Reconstructing the Lecture . . . . .	11
3.3 Chapter Summary . . . . .	14
<b>4 Optical Character Recognition on Projector Systems . . . . .</b>	<b>15</b>
4.1 Capabilities of Optical Character Recognition Systems . . . . .	15
4.2 Difficulties due to Projection Systems . . . . .	17
4.3 Constraints On Data . . . . .	18
4.4 Algorithm For Extraction . . . . .	19
4.4.1 Detect Edges in the Image . . . . .	20
4.4.2 Dilate the Edge Image . . . . .	21

4.4.3	Determine the Connected Components . . . . .	22
4.4.4	Convert the Image to Binary . . . . .	22
4.4.5	Determine Text and Graphic Components . . . . .	24
4.4.6	OCR the Binary Image . . . . .	26
4.5	Results and Additional Figures . . . . .	27
4.5.1	Large Text Image . . . . .	27
4.5.2	Smaller Text . . . . .	28
4.5.3	Poorly Designed Slides . . . . .	28
4.5.4	Problematic OCR Images . . . . .	28
4.6	Chapter Summary and Additional Figures . . . . .	29
<b>5</b>	<b>Speech Recognition of the Lecture . . . . .</b>	<b>39</b>
5.1	Setting Up the Speech Recognition Software . . . . .	39
5.2	Recording the Audio . . . . .	40
5.3	Recognizing the Audio . . . . .	40
5.4	Chapter Summary . . . . .	42
<b>6</b>	<b>Lecturelets: Putting it all Together . . . . .</b>	<b>44</b>
6.1	What is a Lecturelet . . . . .	44
6.2	Getting Lecturelets . . . . .	45
6.3	Parts of the Lecturelet . . . . .	45
6.3.1	Key frame . . . . .	46
6.3.2	Optical Character Recognition . . . . .	46
6.3.3	Speech Recognition . . . . .	47

6.3.4	Additional Components . . . . .	47
6.4	Operations on Lecturelets . . . . .	48
6.5	Implementing Lecturelets and Their Functions . . . . .	50
6.6	Chapter Summary . . . . .	52
<b>7</b>	<b>Conclusions . . . . .</b>	<b>55</b>
	<b>References . . . . .</b>	<b>58</b>



## LIST OF FIGURES

3.1	Four consecutive key frames from a presentation in which the lecture was writing the information to be displayed. Each key frame occurs approximately after one new word was written. . . . .	12
3.2	Four consecutive key frames from a presentation that used a PowerPoint computer generated presentation. Each key frame occurs once a new slide is shown. . . . .	13
4.1	Image projected on the screen by an overhead projector. Note the extreme variance in intensity from the center to the outside corner of the projection. . . . .	18
4.2	Image showing the blank overhead projection with each region pointed out. . . . .	19
4.3	Original overhead image, with text and graphic . . . . .	20
4.4	Edge detected image . . . . .	21
4.5	Dilated Edge Image . . . . .	22
4.6	Dilated edge image with boxes around connected components. This image has 7 blocks. . . . .	23
4.7	Binary Output Image . . . . .	24
4.8	Unformatted text output . . . . .	26

4.9	(Left) An image of an overhead projection (Right) Unformatted OCR output . . . . .	30
4.10	(Left) An image of a computer projection (Right) Unformatted OCR output. In this case, the bullet points was interpreted as an “a” and the letter “I” was added to the final output. The last two words were miss-recognized . . . . .	30
4.11	An image of an overhead projection with complex background (Right) The OCR output . . . . .	31
4.12	(Left)Image of an Overhead projection (Right) Binary output from the system . . . . .	31
4.13	(Left) Image of a computer projection (Right) Binary output of the system. Note that the original image had a marble texture in the background . . . . .	32
4.14	(Left) Image of an overhead projection with a gradient background. (Right) Binary output of the system. Although much of the text was lost, the output image is more readable than the input image	32
4.15	(Left) Overhead image (Right) OCR output . . . . .	33
4.16	(Left) Overhead image (Right) OCR output . . . . .	33
4.17	(Left) Overhead image (Right) OCR output . . . . .	34
4.18	(Left) Overhead image (Right) OCR output . . . . .	34
4.19	(Left) Overhead image (Right) OCR output . . . . .	35
4.20	(Left) Overhead image (Right) OCR output . . . . .	35
4.21	(Left) Overhead image (Right) OCR output . . . . .	36
4.22	(Left) Overhead image (Right) OCR output . . . . .	36

4.23 (Left) Computer Projector image (Right) OCR output . . . . .	37
4.24 (Left)Skewed Slide (Right)Mathematical Symbols . . . . .	37
4.25 (Left) Varying Fonts (Right) Italics . . . . .	38
6.1 Lecturelet containing information about BASIC . . . . .	48
6.2 Lecturelet containing information about Prolog . . . . .	49
6.3 Lecturelet containing information about Java . . . . .	50
6.4 Java program showing text set intersection . . . . .	52
6.5 Java program showing search window, text to search for is “basic”	53
6.6 Java program showing results of search window, and new window ready for next search . . . . .	54

# CHAPTER 1

## Introduction

### 1.1 Statement of Problem

The concept of distance education, or taking classes at a distance has become increasingly popular over the past several years. However, the current methods for distance education cannot keep up with the increasing demand for these classes.

There are primarily two forms of technology based distance education classes. Web based classes and video based classes, both offer advantages and disadvantages. A web-based course is simple to implement, usually requiring at most a basic knowledge of HTML, and the computing power to serve the class. Once a lecture is generated, it simply needs to be placed on the server, and students can almost instantly view the lecture material from anywhere around the world.

There are, however, several disadvantages to a web based course, mostly due to the bandwidth restrictions of the Internet. These bandwidth restrictions make it difficult to present any type of multimedia information, such as video or even detailed images. This will primarily limit a web based course to text only presentation of information and ideas. Even if a university has the bandwidth to support massive amounts of video and other multimedia content, most personal users of the Internet do not have a connection that can support these requirements.

A video based distance education system records a live lecture and distribute the videotapes to distant students. By contrast to the web system, a video based course can do a reasonable job of simulating a classroom environment and display complex multimedia information. However, the cost of the videotapes and their distribution make this an unattractive system as well. Often, in order to cut costs, the videotapes will only be distributed to select central locations. Those central locations will then show the videotape at specific times, requiring students to be at those locations for watching the tape. Finally, the video based system will always have some delay associated with it. A presenter cannot expect his or her students to be familiar with presented material until some fixed time after the lecture, so as to allow time for the tape to arrive at its location.

It is the goal of this thesis to present an implementation of a distance education system that has as many of the advantages of both the web based system and video based system with as few of the disadvantages as possible. The system presented in this thesis is able to take a lecture of any length, and compress the video, first based on key frames (showing new information) and next by extracting text from the video and audio. The end result is a representation of the video lecture that can be several orders of magnitude smaller than the digital form of the entire video.

## **1.2 Audience**

There are primarily three groups that will be interested in this thesis: educators, computer scientists, and students. Educators will be interested in a new distance education method, and the means of transmitting course content over the Inter-

net. As with other Internet based distance education systems, this system will cut the transmission time from days or weeks (as with a mail based systems) to minutes or hours, depending on the complexity of lecture to be transmitted. This is important when teaching to a large class at a distance, as it becomes easier for the instructor to keep students in remote areas from falling behind. This system is also of interest to educators in that it is an inexpensive system to set up and administer, whereas video delivery systems require producing and distributing videotapes, which can be very costly. This system integrates well with the current equipment of video systems, and generally requires only a few additional hardware components to existing computers.

The computer science community will find interest in this thesis for several reasons, from both the problems addressed, and other interesting questions that arise from the thesis. From a computer science perspective, this thesis addresses problems from several areas. In computer vision, techniques are used to process images for optical character recognition (OCR), and find key frame images in a lecture. Networking becomes an issue when the most efficient algorithms in order to transfer the lecture video is considered. The transfer, compression, and other techniques are of interest to those involved in multimedia. Discrete mathematics and set theory are also used to a certain extent, as discussed later in this thesis.

Finally, students will have an interest in this thesis, as the results can ultimately affect the way that they will take courses. Students who wish to participate in distance education courses need to be aware of the latest systems and technologies that are being used. Their input and feedback is critical to successful, widespread use of such systems. If implemented on a large scale, the system described should be of great interest to students.

### 1.3 Structure of the Thesis

This thesis describes a system to implement a video based on-line distance education system. In order to overcome the bandwidth restrictions, several methods will be presented to extract information from a videotaped lecture. The process of extracting this information will result in a compressed version of the video, while not losing any of the classroom information. The expectation is that the lecture will be centered around some type of projection system, such as an overhead projector or computer generated PowerPoint presentation.

First, a method for extracting “key frames” from the video will be discussed. These key frames represent the visual state of the lecture at any given time. A new key frame is generated anytime new visual information appears in the video. When displayed in sequence for the correct amount of time, synchronized along with the audio of the lecture, a compressed version of the lecture is shown.

Next, processing is performed on each of the key frames, so that it may then be run through an optical character recognition program (OCR). In doing this, both the text and graphics of the key frames will be extracted. This way, the entire text of a presentation used during a lecture can be captured and stored for later analysis, search and indexing of the lecture.

At the same time the OCR processing is going on, the audio of the lecture is also analyzed. The audio is processed through a speech recognition program, which will create a set of text that is spoken during the lecture. The output of this process, like the OCR output, can then be used to analyze, search and index the lecture.

Finally, the idea of a “lecturelet” will be introduced. The lecturelet, a new concept in distance education, represents a small discrete unit of lecture. The

information extracted in the above methods represent the various components of a lecturelet. The lecturelet provides a simple means of performing operations on the lecture, which will ultimately make the lecture more interactive than a simple videotape would provide.



## CHAPTER 2

### Previous Work

The computer's ability to interpret lectures has become a topic of research with the explosion in multimedia technology. Recent work [3] uses gesture tracking and changes in background to index a video of a lecture involving overheads. Gesture tracking is used to determine the importance of any given part of an overhead projection, and changes in the background aid in figuring out when the overhead has changed [3]. This work is intended to permit using the information contained on the overhead projection in order to index the lecture. The work [3] is not able to understand or process the contents of the overhead slides.

There have been other methods proposed to extract text and graphics and prepare images for Optical Character Recognition [4] or other types of document analysis. The projectors that have been focused on in this work have a unique lighting setup (discussed later in this paper), which can cause problems with the method's such as the ones proposed in [4, 5]. Ultimately a system could be developed which uses this system, in conjunction with others to automatically extract all information from videotaped lectures.

Distance learning has increased over the past several years. With new multimedia and web based technologies it is becoming easier to offer courses over the Internet [8]. Programs such as WebCT [9] allow students and professors to communicate with each other, and the entire class as a whole, in a simulated

classroom environment. However, all discussions and lectures are limited to text based, mostly due to the required transfer sizes of sound and video files. In order for a videotaped lecture to be downloaded in a fast enough manner, the frames would need to be small, and compressed. Both size and compression would contribute to the inability to read detailed information from the projector.

Other systems similar to the one described in this thesis have also been introduced, such as the CueVideo System [2] at IBM. This system is intended to record and archive lectures, and segment these lectures based on overhead projections. Unlike the system described in this thesis, however, CueVideo requires much more computer hardware, and processing time, in order to get similar output to the system described here. Additionally, presenters are required to “leave behind” a copy of their presentation for analysis. Often, presenters will not be willing to do this, especially when the presentation slides are printed on expensive plastic film. This system does not have any of these requirements.

## CHAPTER 3

### Extracting Key frames from a Videotaped Lecture

Whenever a classroom lecture is videotaped for distance education purposes, the camera set up is usually either one of two ways. The first is that static camera (or single view) will be set up in the back of the classroom, and face the board, or projection screen (depending on the lecturer's presentation style). The second option is to set up several cameras or camera views, that can be changing through the entire lecture. Generally these views will include the board/projection screen, the lecturer, and perhaps a view of the students in the classroom. In either situation, most of the recorded information will be redundant. One video based distance education system, FEEDS [12], uses this classroom setup. In terms of digital representation, storage and transmission of video can range between 12 to 30 frames per second. Reducing the frame rate will cause the video to become choppy. If the frame rate is reduced enough (one frame every several seconds), the video will begin to look like a slideshow. Either way this is unacceptable for television and movies, where the smooth motion is essential to the enjoyment of the show. However, in the case of a classroom, the material being presented is more important than smooth motion. During a PowerPoint presentation, a reduced frame rate will not even be noticeable; a PowerPoint presentation is in fact itself a slideshow. If the lecturer aims the camera at him or herself, then

a single image representing this shot sequence will suffice rather than capturing the entire motion of the lecturer.

This chapter describes a method for reducing the frames in the video to only those containing new information (key frames). These key frames represent a compressed version of the lecture. By displaying each of the key frames in their order of appearance for the amount of time that they appeared, this can reconstruct the entire lecture at a much smaller storage and transmission cost. Key frames also serve to naturally partition and index a lecture. In addition, key frames can be used for further analysis, which will be discussed in later chapters.

### **3.1 Extracting the Key frames**

In order to extract the key frames, the system assumes that there is some video camera pointed towards a projection screen, with no obstructions between the screen and the camera. There is no restriction on what is generating the projection (i.e., an overhead projector, slide projector, computer projector, etc.). This is the exact setup of the FEEDS classrooms as discussed above.

Since the information contained in a lecture is not changing rapidly, and many times not at all (such as a videotaped PowerPoint Presentation, where the same slide can be displayed for several minutes), it is not necessary to transmit or store 30 frames per second. Instead, the lecture is compressed by use of key frames. In this context, a key frame is defined to be the first frame of video containing new information. Some examples of key frames in a computer generated presentation are a new line of text, a new graphic or a new “slide” altogether. In the case of a lecture where the presenter is writing and that writing is in turn displayed on

the screen, a key frame is when new information is added to the projection or when the presenter points to a particular area of the projection. The following is the algorithm that is used to determine if a frame is a key frame or not:

1. The first frame is a key frame
2. While receiving video input do
  - (a) perform difference operation between the previous frame and current frame
  - (b) if difference operation returns true, perform difference operation on current frame and most recent key frame
  - (c) if second difference operation returns true, found key frame, record time that key frame is detected

The difference operation is as follows:

$$PicDiff = \frac{\sum_{i=1}^m \sum_{j=1}^n PicDiffIsBig(Pic1_{ij} - Pic2_{ij})}{m \times n} \times 100$$

Where  $Pic1$  and  $Pic2$  are the two images in question,  $m$  and  $n$  are the dimensions of the pictures.  $PicDiffIsBig$  is a predicate that is 1 iff

$$|Pic1_{ij} - Pic2_{ij}| \leq tol$$

where  $tol$  is some number chosen close to zero. The difference operation returns true if  $PicDiff$  is greater than 7.5, which has been experimentally found to represent significant change.

This algorithm works extremely well for all types of projections. In the case of a non-computer generated presentation, the lecturer may point to a certain region of the projection; depending on the amount of motion, this may or may not be

detected as a key frame. Other cases of noticeable loss of classroom information has not been found.

Since this step requires the computer to keep only three frames in memory, and this process runs quickly, it can be performed as an online operation. The video signal simply needs to be connected to the computer, and at the conclusion of the lecture the key frames have been extracted. In order to replay the lecture, the key frames are displayed in correct order and for the correct amount of time, along with the audio from the lecture. The compression of the lecture depends on how many key frames were generated. On average, a 1 hour lecture was about 50 megabytes, however this varied from 20 to 80 megabytes, with most of the storage cost being the audio.

Figure 3.1 shows four consecutive key frames that were extracted from a lecture in which the presenter was writing the information that was being displayed on the projector screen. A new key frame is extracted approximately after each word is written. Figure 3.2 shows four consecutive key frames in a videotaped lecture where the presenter used a PowerPoint computer generated presentation. A key frame is extracted each time the presenter shows a new slide.

## **3.2 Reconstructing the Lecture**

Each key frame represents a change in the visual state of the lecture. By displaying each key frame in the order of appearance for the correct amount of time, the entire lecture can be reconstructed with much compression. In order to make the lecture viewable on-line, this system uses the RealNetworks program Real Slideshow. The input to Real Slideshow is a set of images and a sound file. The

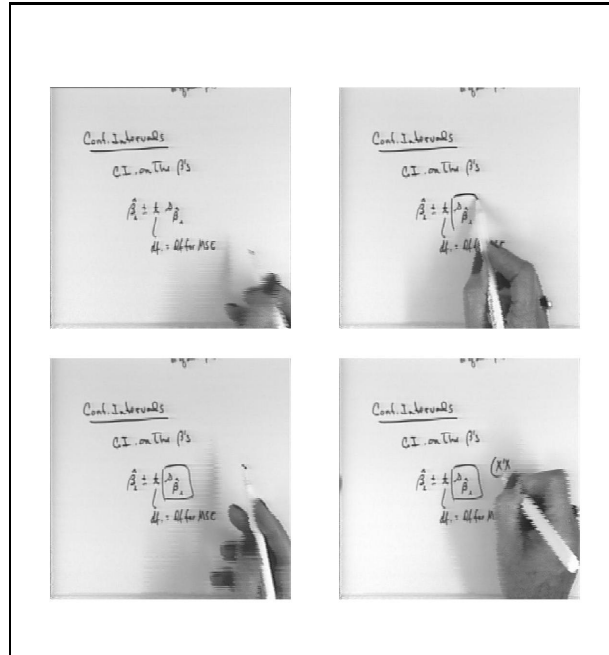


Figure 3.1: Four consecutive key frames from a presentation in which the lecture was writing the information to be displayed. Each key frame occurs approximately after one new word was written.

program will output a “RealMedia” presentation, in slideshow form. In order to convert a key framed lecture into slideshow presentation, all the key frames are placed as input to the program, in addition to a digital recording of the audio from the lecture. Additional programs were written to interact with the output of the Real Slideshow program and automatically set the timing for each key frame, so that the video and audio remain in synchronization with each other.

The RealMedia output file can be viewed from any personal computer which has a sound card and RealPlayer installed. There are free versions of the RealPlayer software for every popular operating system (Windows, Macintosh, UNIX, etc.), so very few students would have difficulty viewing the lecture. The RealMedia file format is also set up to be streamed over the Internet, so a RealMedia Internet Server can be set up in order to send the lectures to distant students.

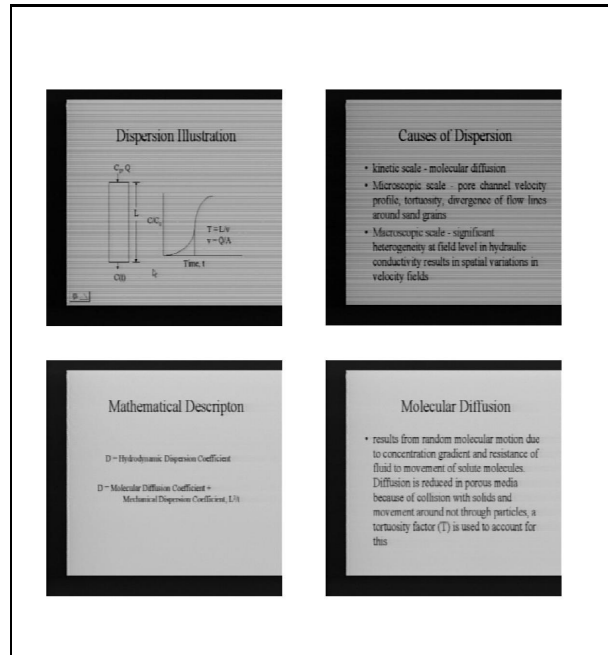


Figure 3.2: Four consecutive key frames from a presentation that used a Power-Point computer generated presentation. Each key frame occurs once a new slide is shown.

Because the lectures are being transmitted over the Internet, each lecture can be viewed by the students within a couple of hours after the lecture has ended. This would take several days or weeks if the videotape had to be mailed to each student taking the course.

The method of capturing the key frames, and converting it to a RealMedia presentation was preformed on a FEEDS course for one semester. The professor for the class used a document camera to present the material. In this setup, he wrote on a sheet of paper, which was in turn projected onto a screen by a computer projector. The average class period generated between two and three hundred key frames, and would take approximately 40 megabytes of space. Using this system, the entire semester's lectures were able to be stored in under one gigabyte, or on 3 standard CDs.



### 3.3 Chapter Summary

This chapter presented a method for extracting key frames from a videotaped lecture. A key frame is an image which represents new information that is being presented during the lecture. The key frames represent a natural compression and partition of the lecture, reducing the number of frames that need to be stored by several orders of magnitude. After capturing the key frames, the lecture can be converted to a web ready slideshow presentation by using RealNetworks products, including Real Slideshow and RealPlayer. Further operations can be performed on the key frames, including Optical Character Recognition, and information mining, as described in later chapters.

## CHAPTER 4

# Optical Character Recognition on Projector Systems

Overhead and computer projectors in the classroom and presentations have become very popular in the past several years and are used in just about every lecture or presentation today [7]. In this chapter, methods are presented for taking a single image which contains some type of projection (overhead, computer, slide, etc.) and extract the text and graphics from the image.

The key frames that were found in the video are used as input to the algorithms that are presented here. By extracting the text and graphics from the key frames, the images can be even further compressed down to small graphics and ASCII text. This output text can then be used for indexing and segmenting the video, as well setting up a means of searching through the lecture.

### 4.1 Capabilities of Optical Character Recognition Systems

Methods presented in this chapter use Optical Character Recognition (OCR) to extract information from the key frames. The algorithms used for OCR are

well developed and continue to improve. Several OCR programs are available commercially. However, the algorithms require very constrained data. Most OCR programs will only work on images of documents (scanned images). The programs begin to break down as soon as extraneous information is introduced to the image. Some examples are staples or creases in the original images, smudges from ink, and second or third (or later) generation photocopies. Each of these issues and many others will cause the recognition rate to decrease. Images that are not of documents will almost always cause OCR programs to fail. Consider an image of a billboard on the side of the road. Although the billboard itself does contain text, the varying colors and shades of the images, the graphics on the billboard and the changing font types and sizes will all contribute to the breakdown of the algorithm. Other background "noise" also contributes to the overall failure of the algorithm.

Several OCR programs on the market today are able to recognize text by use of Principle Component Analysis or some similar recognition means [6]. Most OCR programs come with a preset database of characters, and require training to be able to recognize additional characters and fonts. Training is essential when dealing with character recognition of handwriting, since no two people have identical handwriting. When OCR technology is able to accurately read handwriting, this method will properly extract text that is either type or hand written; however, OCR technology is still currently in its nascent stages for recognizing handwriting. Once the technology is capable of handling handwriting, the methods presented in this chapter will still be valid, and be able to work with the newer OCR systems.

## 4.2 Difficulties due to Projection Systems

The design of projectors introduce problems in terms of being able to extract information from the projector. Projectors are built with an intense single bulb below a clear glass stage. Because of the single concentrated light source, the pixels in the center of the projection are extremely bright, while the further from the center, the darker the pixels become. This rapid change in the lighting will cause an unprocessed image to be unrecognizable by the OCR program. This lighting variance is shown below in figure 4.1. A similar design is used for other projectors, which causes the same problems.

One way of viewing a projector system is in the following way. An image that contains a projection has seven regions. The first region is the part of the image that does not contain the projection at all. The next four regions are the 4 corners of the projection. The corners are going to be the darkest region that may still contain text or graphic information. Moving towards the center of the projection, there is a ring region that is brighter than the corners. The final region is the center of the projection. This is the brightest part of the image. There is so much variation in the lighting that a dark text or graphic pixel appearing in center (brightest) region will be brighter than a non-text or non-graphic region in a corner of the image. The result of this is that when standard single threshold segmentation algorithms are used the corners of the projection are almost always seen as text/graphics and the center is never seen as such. This presents problems which are addressed later in the paper.

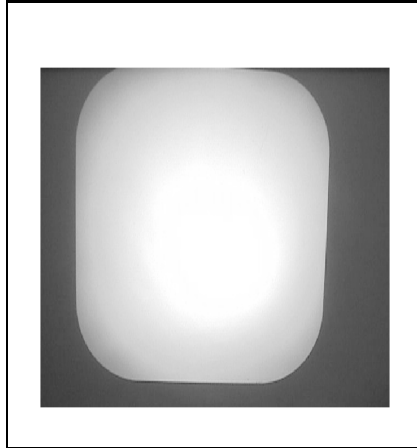


Figure 4.1: Image projected on the screen by an overhead projector. Note the extreme variance in intensity from the center to the outside corner of the projection.

### 4.3 Constraints On Data

Although the method presented in this chapter will work on images acquired at any resolution, the OCR programs have very strict requirements. The algorithms for optical character recognition are designed for scanned documents, which are generally of 300 dpi (dots per inch) resolution. The OCR program will not properly recognize lower-resolution images, as acquired by standard video camera/capture card combinations. One way around this is to have very large font sizes (such as 50+ point fonts) to compensate for the low resolution. This does not happen often in real life, so it is best to capture images at the highest resolution possible.

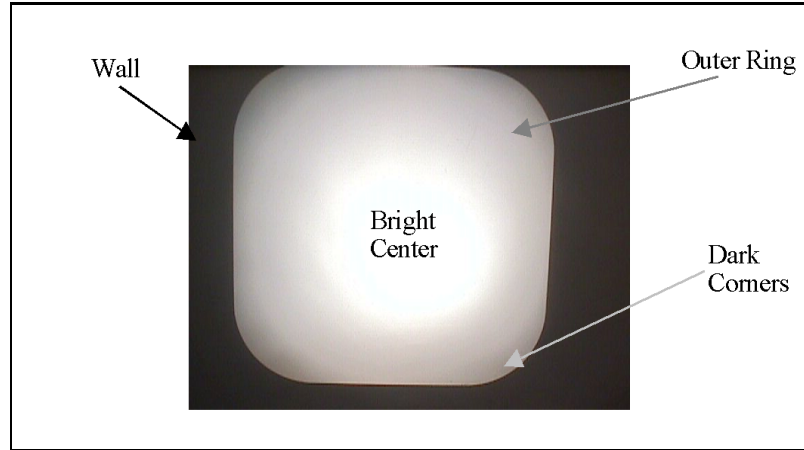


Figure 4.2: Image showing the blank overhead projection with each region pointed out.

#### 4.4 Algorithm For Extraction

Several steps of this method is similar to other work [4], as they considered a related problem: that of text extraction from scenes. Here the overall algorithm for text and graphic extraction is presented. After that, details of each step are shown, as well as differences between this and other methods. Figure 4.3 shows the input image of an overhead projection.

1. Detect edges in image
2. Dilate edge image
3. Determine connected components
4. Convert image to binary
5. Determine text or graphic components
6. Process image



Figure 4.3: Original overhead image, with text and graphic

#### 4.4.1 Detect Edges in the Image

Text and graphics have a high amount of contrast. If they did not have a high contrast to surrounding areas, they would not be visible. By extracting the edges from the image, an outline of all of the text in the image is obtained, as well as the outline of any clipart and the border of the projected image. Taking the edge will also help in the case of images that have some sort of complex background. A slow gradient in the background exists from the lighting in all projections, as shown in figure 4.1 and is a common effect in presentation slides. Since the background is changing slowly, there will not be enough contrast to constitute an edge, and ultimately, such a background will be ignored.

For this work, a Sobel edge detection algorithm is used, with two possible preset thresholds, to get a binary output. Computer projected images are much darker than [plastic] images projected by an overhead projector. Since the computer images are darker, they have a more compressed and shifted histogram. This fact is used to determine the type of image shown and which of the two preset thresholds to use on any given image.

Figure 4.4 shows the edge image, notice that the background variance (in 4.3) is now ignored.



Figure 4.4: Edge detected image

#### 4.4.2 Dilate the Edge Image

After extracting the edges from the image, dilate each edge. This helps in three ways. First, if any of the edges become broken during the previous step, the dilation will reconnect those edges. Second, by expanding the edges, characters such as the letter “i” will become completely connected. Finally, the dilation will cause all of the letters in each word to become connected to each other, but not to surrounding words. These connected edges is useful in the connected components step of the process, described in the next section of this chapter. Figure 4.5 shows an example of the dilated edge image.





Figure 4.5: Dilated Edge Image

### 4.4.3 Determine the Connected Components

A simple connected components algorithm is applied to the dilated edge image. This will cause each word or graphic in the image to be seen as a single region, or block. Additionally, the outline of the projection, and any noise pixels that were included in the edge image will be labeled as a connected component. Since noise is always small, it can be immediately removed on the basis of size. Likewise, the outline of the projection will also be removed, under the assumption that no reasonable block can take up such a large amount of the projection. Figure 4.6 shows each individual block in an image containing 7 blocks.

### 4.4.4 Convert the Image to Binary

Before the OCR program or other recognition algorithms can be used on the image, the information (text and graphics) must be segmented from the background and the entire image must be converted to binary. Binary is meant to

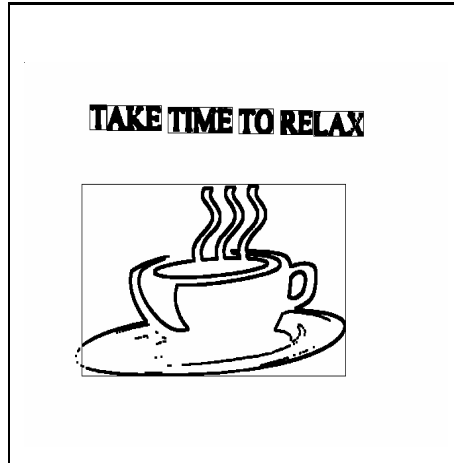


Figure 4.6: Dilated edge image with boxes around connected components. This image has 7 blocks.

mean a two-state (black or white) image. The rapidly changing light conditions (as alluded to in the discussion related to Figure 4.1) cause many problems for segmentation. Most segmentation techniques are based on histogram models of the entire image, or of a large area [5]. In the image sets used for this chapter, a large area will have a lot of lighting change, and the segmentation would fail. A small area can not be used either, because histograms are based on statistics. If the area is too small, the statistics function will not be accurate and again, the segmentation would not work properly.

In order to create a binary image, the following method is adopted. Each pixel in a block has a small mask built around it. If the pixel in question is significantly darker than its neighbor pixels in the mask, then it is marked as text (black), otherwise it is marked as background (white). (If the standard overhead image was inverted, to have a dark background on light text, then each pixel would simply be checked to be much brighter than it's neighbors.) A mask size of  $20 \times 20$  pixels was found to be a reasonable choice for all the images that

were tested. While this method is computationally expensive, it has been found to be the best way to cope with the problem of varying light.

Figure 4.7 shows the image after it has been converted to binary. Parts of the cup in the original image did not pass the segmentation test. However, when that block is tested to see if it is text or graphic in the next step, it is found to be a graphic, so the original (non-segmented) graphic can be it is recovered. Many other examples are shown in the following sections.

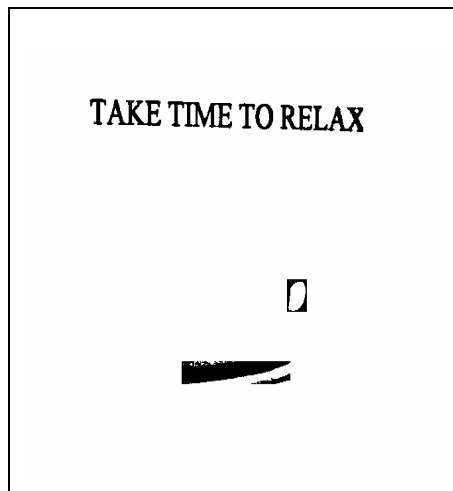


Figure 4.7: Binary Output Image

#### 4.4.5 Determine Text and Graphic Components

After finding connected components of each region and segmenting these regions, it is necessary to separate the graphic regions from the text regions. For this step, Optical Character Recognition program output determine the results. The OCR program is used in the following algorithm:

1. Use the OCR program to analyze each of the blocks separately.

2. Check each block against the “text rules” (discussed below).
3. For each block, if it has violated a text rule, then it is marked as a graphic, else it is text

Once recognized by the OCR program, text will display different characteristics from graphics. Even in the event that a word is misinterpreted by the OCR program, that word will still maintain its “word” characteristics. Considering these characteristics, the following rules have been defined for the OCR output of each block. A block that violates any of these rules is marked as a graphic:

1. Some text information must be returned.
2. Only one line can be returned.
3. Returned characters must be within the range of 32 to 166 on the ASCII chart.

The justification for each of the three rules are as follows. If there was no information returned (Rule 1) then the OCR program was unable to read any part of the block, and returned the same block as a graphic, which would not be present in an ASCII file. Since each text block should only consist of one word, if more than one line is returned (Rule 2) then more than one “word” was contained in the block. The block must be a graphic. Finally, since the OCR program attempts to match to any character in the extended ASCII chart, acceptable range of characters that can appear in a projected image (Rule 3) has been determined. Anything appearing out the range is most likely a part of a graphic. Any block that violates a rule is marked as being a graphic.

#### 4.4.6 OCR the Binary Image

Once the image has been run through the first 5 steps, it is then ready for the optical character recognition processing. It is important to note that this is separate from the OCR processing that was performed in step 5. Any “off the shelf” OCR program should work fine here. The details of the OCR program that is used are discussed in the next section of this chapter. Figure 4.8 shows unformatted text output of the OCR program, the graphics have been removed from the image. The system is aware of graphic blocks, and they can be processed by a different method separately. This awareness also compensates for the missing part of the graphic in the previous step. Afterwards, the text can be run through a spell check program to correct minor errors in recognition.



Figure 4.8: Unformatted text output

## 4.5 Results and Additional Figures

For the tests, Caere OmniPage Pro (version 9) was used to “read” the binary images, and to create the ASCII files to use in determining if the block is a word or a graphic. The method can be tested with or without the graphic extraction algorithm. All of the images that were tested the graphics extraction on were successful in determining the graphic blocks from the test blocks. Therefore, in this section only the text extraction is discussed.

Since the OCR program expects images to be scanned, images captured by a video camera will generally not have a high enough resolution to be properly recognized. To correct for this, and to test the method’s viability, mostly images with a large font size were intentionally used. The results are broken down into several subsections. Different criteria for “success” are used for each section. Those criteria are discussed accordingly.

### 4.5.1 Large Text Image

The first group is large text (50 to 60-point text). This was to correct for the low resolution of the digital image. A success in this group is a character or word that is correctly recognized by the OCR program. All of the text in the output images (step 5) in this group were completely readable by humans. The overall success rate of the OCR program was approximately 95%. Figures 4.19 through 4.22 show several examples of pictures from this group.

### 4.5.2 Smaller Text

In the second group, the text is considerably smaller than the text used in Group 1 (approximately 36 to 44-point font size). While the presented method is able to extract the text from images of this type, it is here the OCR algorithms begin to break down. Comparing figures from the previous group and those in this group show a sharp reduction in the recognition. The binary image in this section is still readable. Figures 4.9 to 4.17 (excluding 4.14) show smaller text figures.

### 4.5.3 Poorly Designed Slides

This is the set of images that could not be processed by the presented method. They consisted of images where the text was too small, or there was not enough contrast between the text and background. All of the projections in this group were not readable, and each image violated one or more of the assumptions stated at the beginning of this chapter. Because these are ill-suited images, there was no way to classify a successful test. Figure 4.14 shows an image with a large gradient. It is important to note, that although this image is not readable (by humans or OCR), the binary output is more readable by humans than the original image.

### 4.5.4 Problematic OCR Images

While this method makes no assumptions about the type of text, the OCR algorithms are not as forgiving. This problematic group of image includes italic text, mathematical symbols, varying symbols, a skewed image (either the plastic

slide or the camera was skewed), and other similar events. The method is able to extract the text from the image, however, the OCR programs are not able to correctly recognize the text. Examples of these images are shown in figures 4.24 and 4.25. It should be clear that as OCR technology improves to be able to handle this group of images, they can be processed by this method as well.

## 4.6 Chapter Summary and Additional Figures

This chapter looked at ways in which videos of lectures using overhead and computer generated projections can be analyzed by a computer and more specifically, have the text of the projection recognized by an OCR program and the graphics extracted (for other use). This processing can be used to aid in distance learning, make a video lecture searchable by keywords in the overheads, and many other ways not discussed. If a web broadcast is compressed, information on an overhead projector will often be lost. If the projection is analyzed separately, the video can be compressed and the information on projected image can be transmitted apart from the video.

Several data sets were shown, demonstrating the methods capabilities. A discussion of all examples was presented as well. Negative examples, which causes the system, OCR component or both to fail, were also shown. Some examples of the OCR program failing were italic text, mathematical symbols, and varying font types. Failures in the framework presented were in cases in which the text did not have enough contrast to be segmented from the background, such as in a gradient image. Positive examples, which follow the constraints of the system, had very high accuracy (80% - 100%).



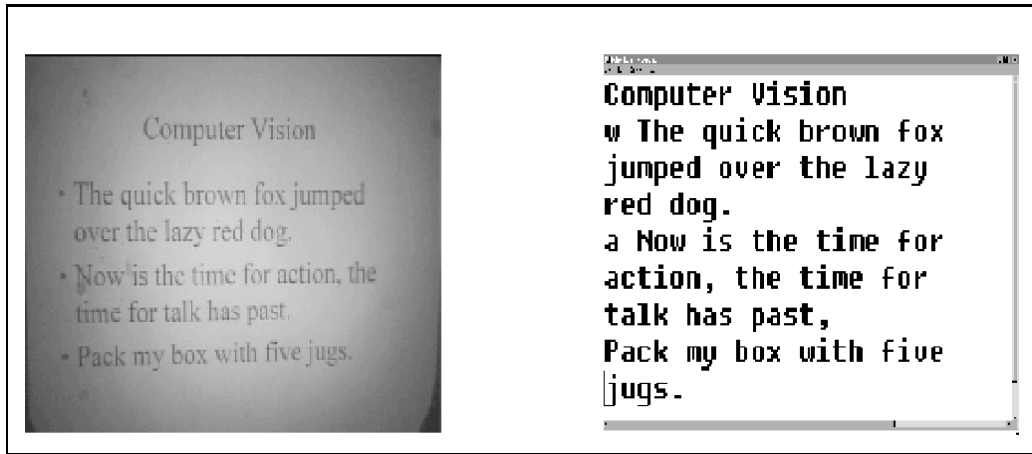


Figure 4.9: (Left) An image of an overhead projection (Right) Unformatted OCR output

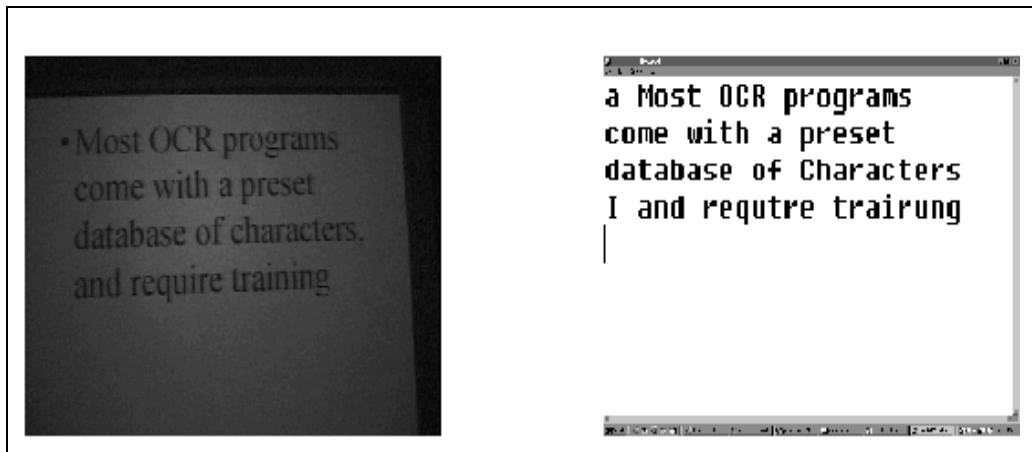


Figure 4.10: (Left) An image of a computer projection (Right) Unformatted OCR output. In this case, the bullet points was interpreted as an “a” and the letter “I” was added to the final output. The last two words were miss-recognized

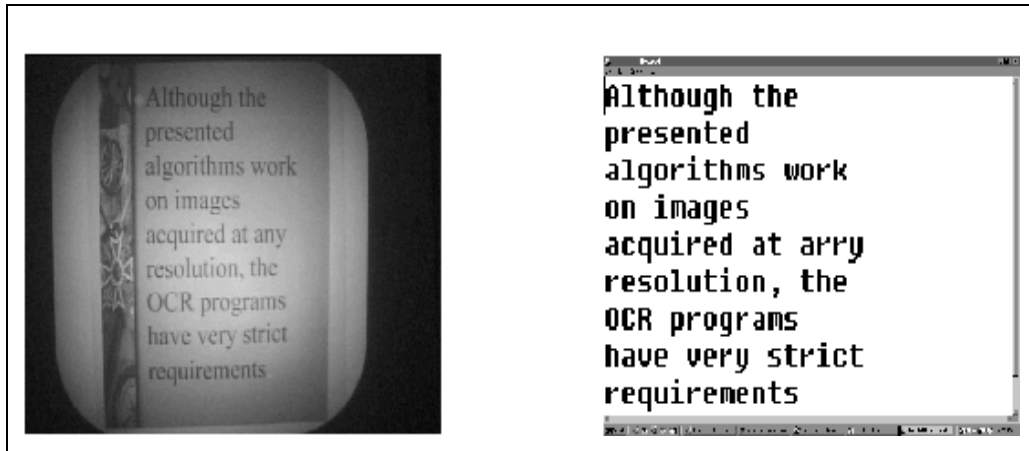


Figure 4.11: An image of an overhead projection with complex background (Right) The OCR output

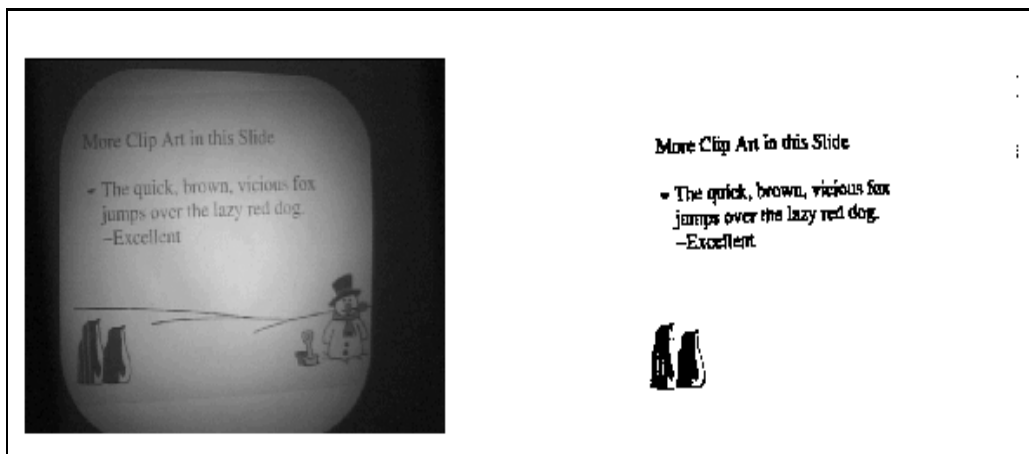


Figure 4.12: (Left)Image of an Overhead projection (Right) Binary output from the system

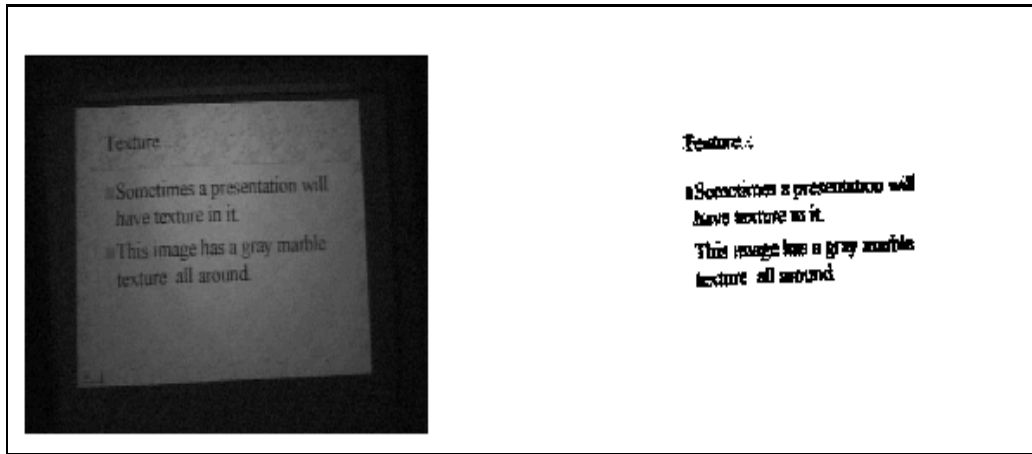


Figure 4.13: (Left) Image of a computer projection (Right) Binary output of the system. Note that the original image had a marble texture in the background

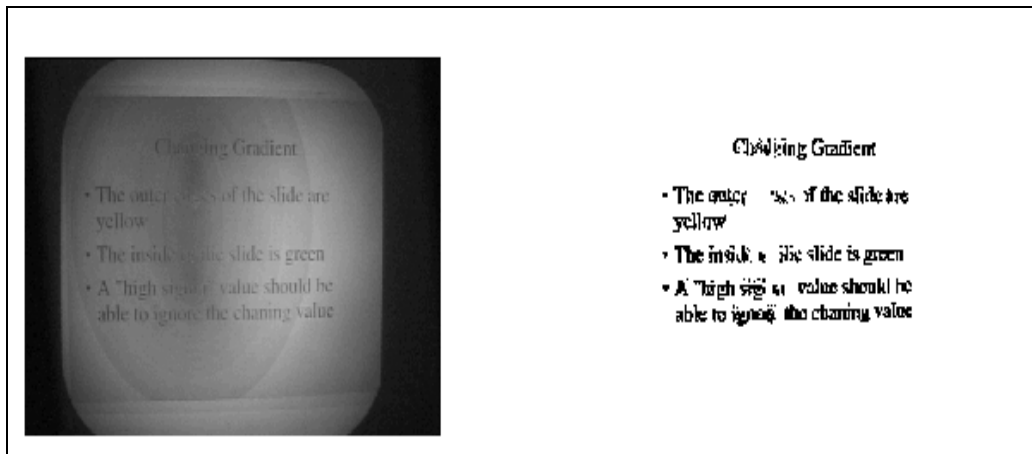


Figure 4.14: (Left) Image of an overhead projection with a gradient background. (Right) Binary output of the system. Although much of the text was lost, the output image is more readable than the input image

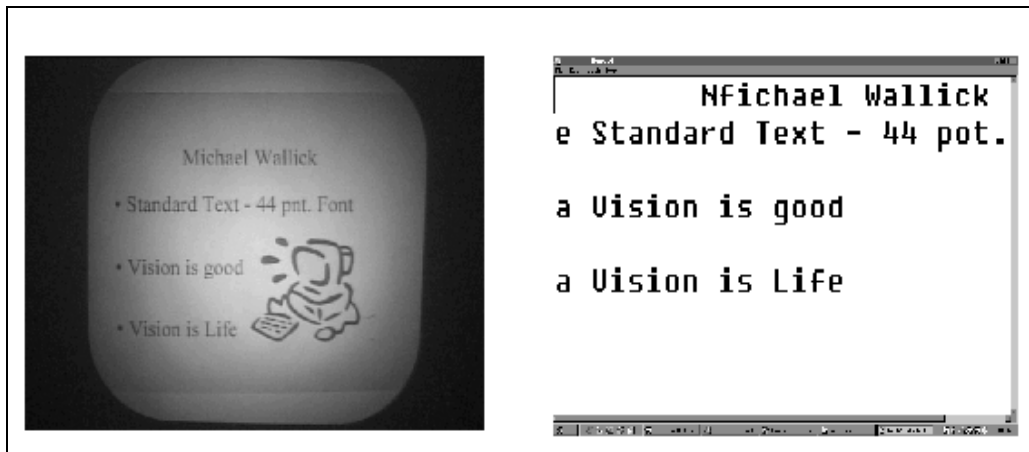


Figure 4.15: (Left) Overhead image (Right) OCR output

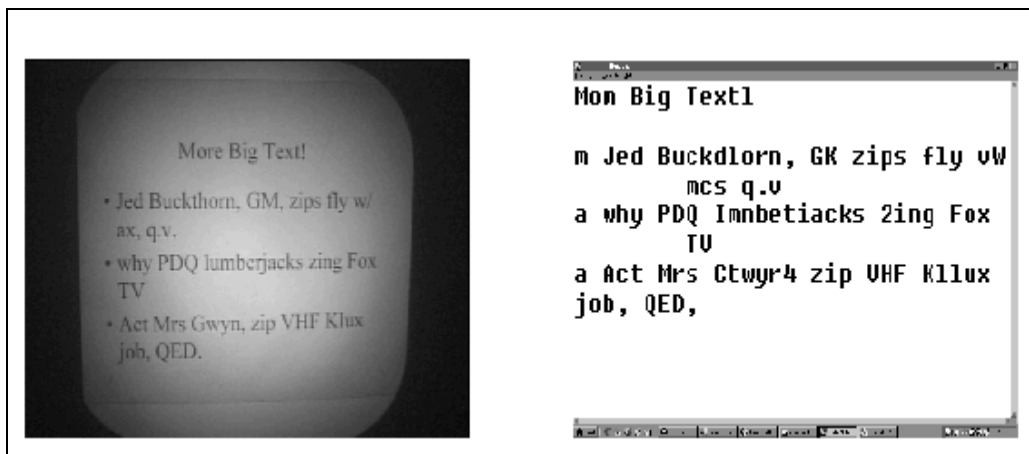


Figure 4.16: (Left) Overhead image (Right) OCR output

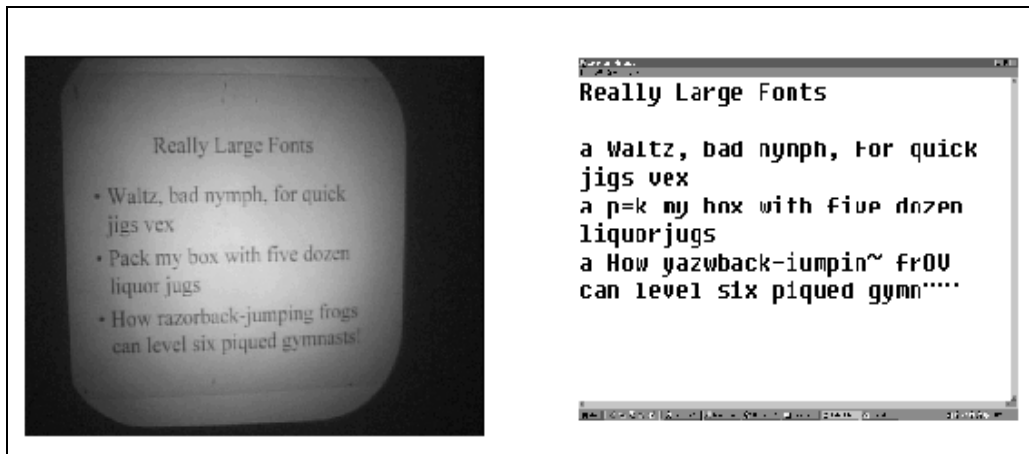


Figure 4.17: (Left) Overhead image (Right) OCR output

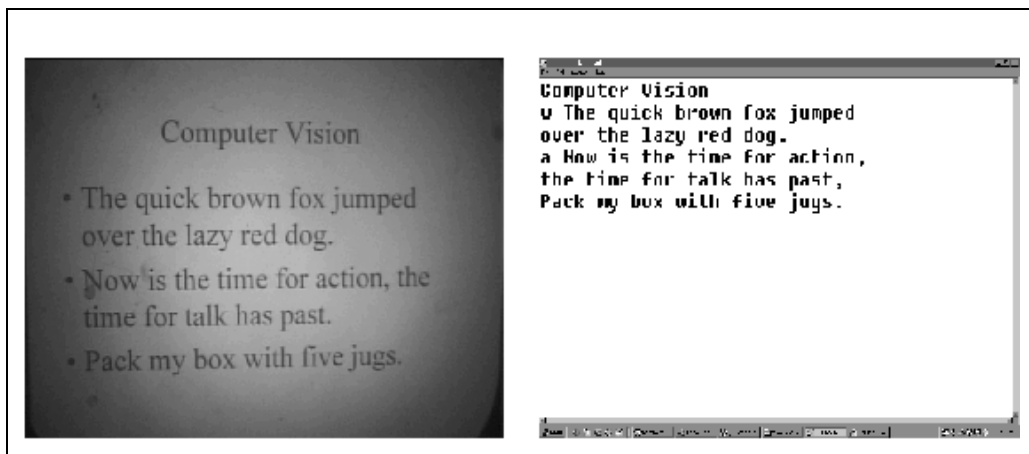


Figure 4.18: (Left) Overhead image (Right) OCR output

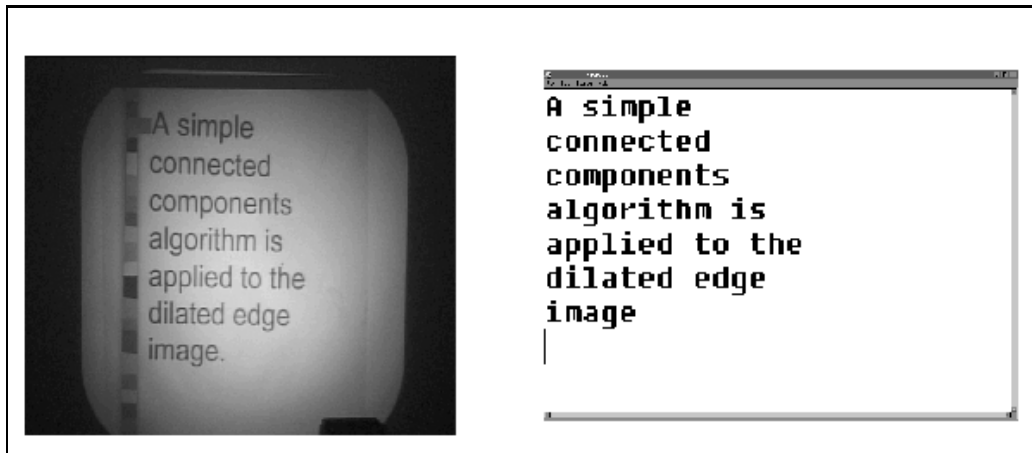


Figure 4.19: (Left) Overhead image (Right) OCR output

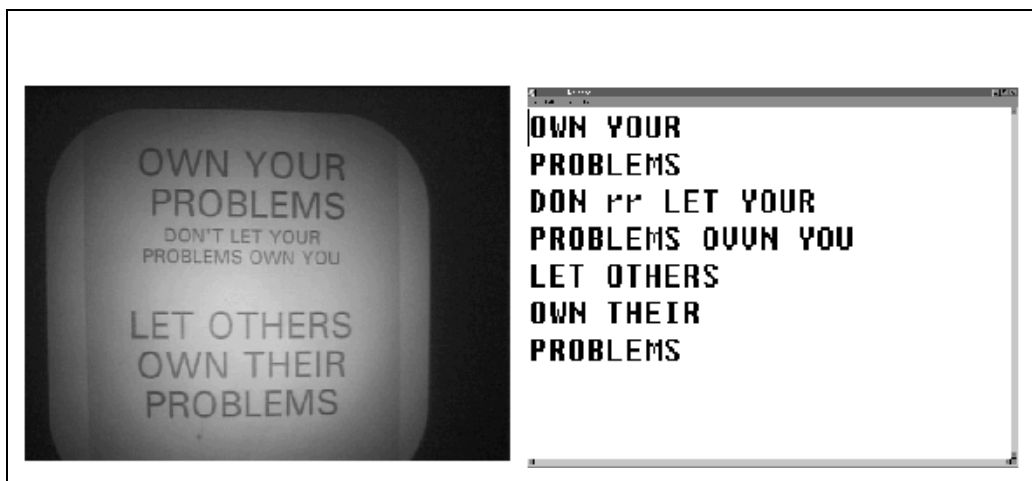


Figure 4.20: (Left) Overhead image (Right) OCR output

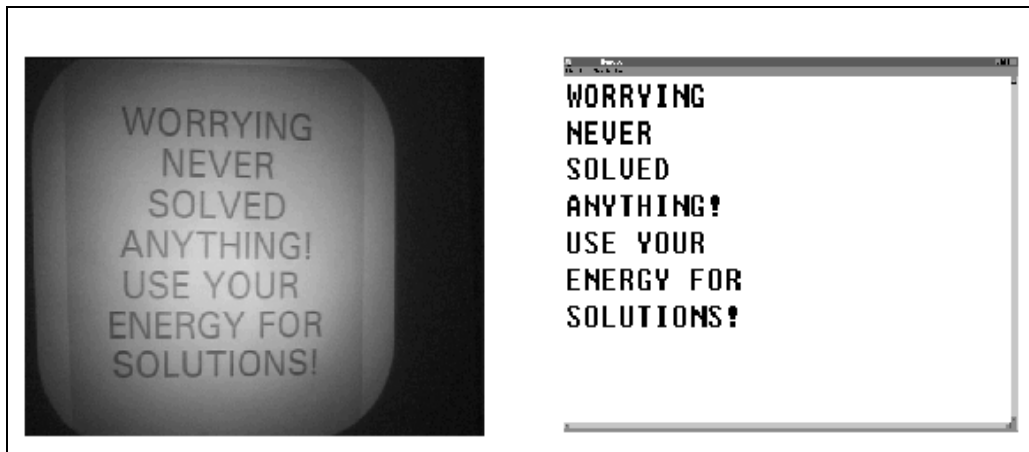


Figure 4.21: (Left) Overhead image (Right) OCR output

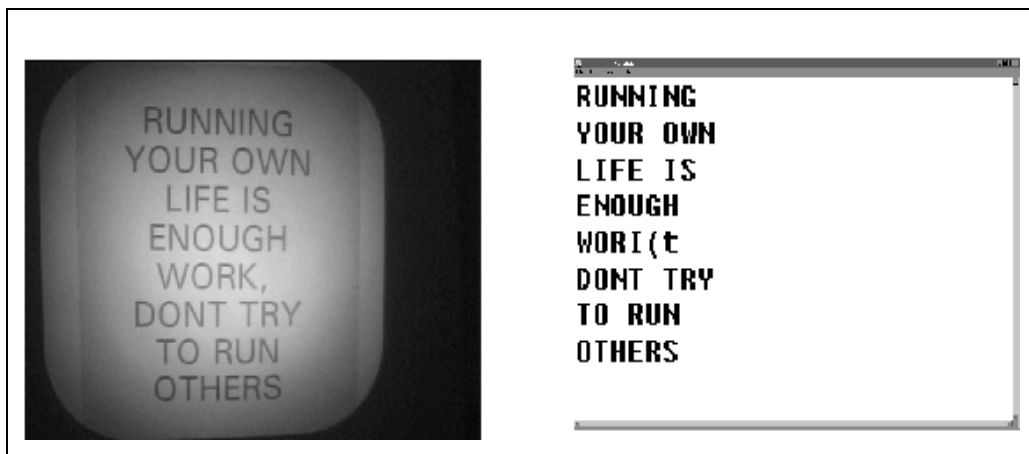


Figure 4.22: (Left) Overhead image (Right) OCR output

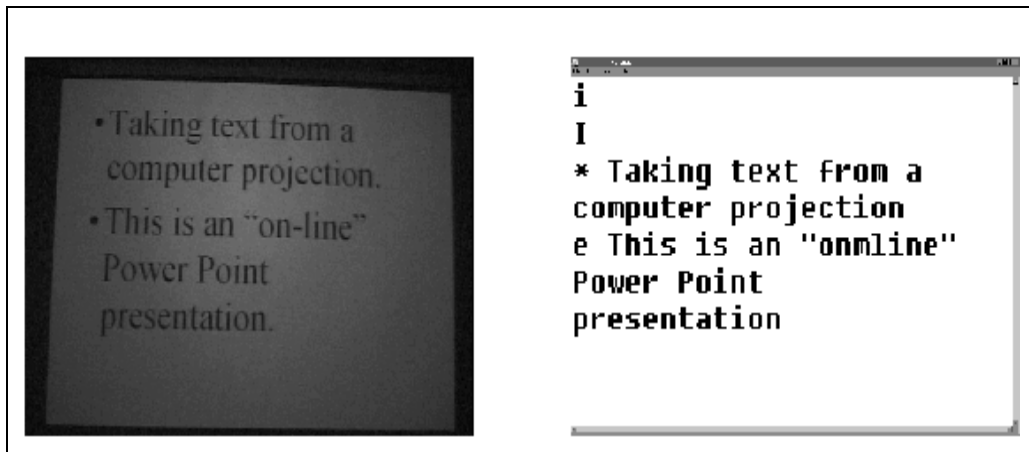


Figure 4.23: (Left) Computer Projector image (Right) OCR output

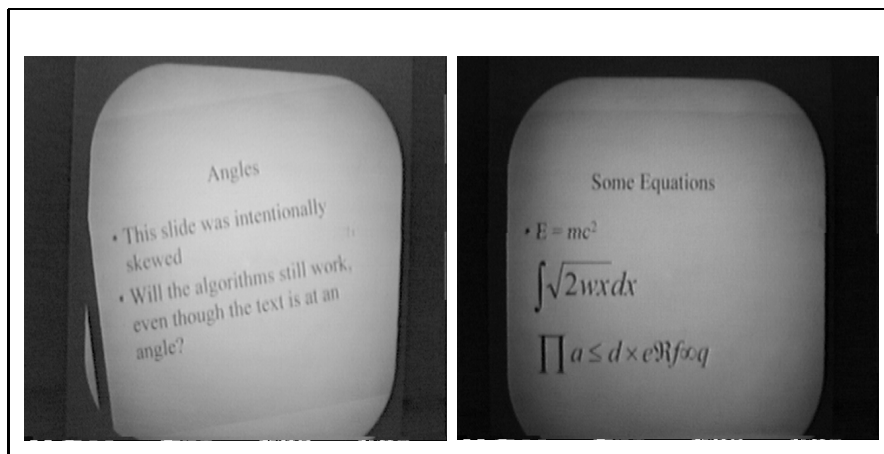


Figure 4.24: (Left) Skewed Slide (Right) Mathematical Symbols



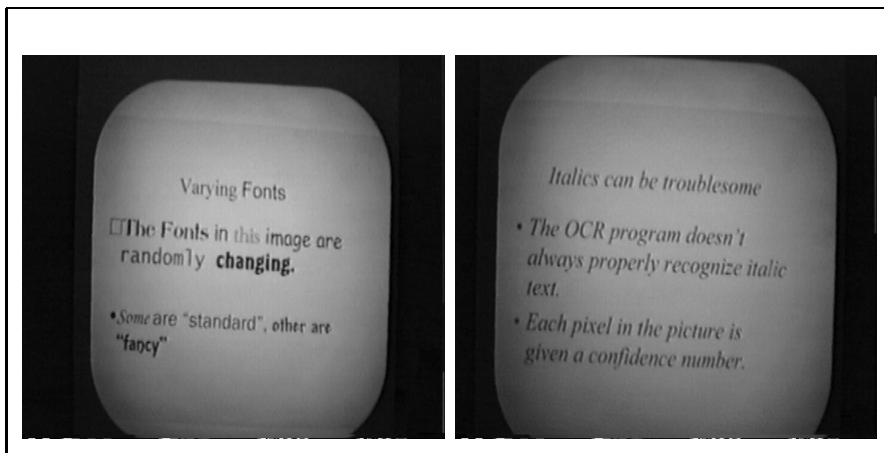


Figure 4.25: (Left) Varying Fonts (Right) Italics

## CHAPTER 5

### Speech Recognition of the Lecture

Extracting the text from the projector is not the only form of data mining that can be performed on the lecture. Another powerful method of extracting information is by speech recognition. Speech recognition is converting the audio of the lecture into a text file, such as an ASCII document. There have been many recent advances in Speech Recognition technology that makes this idea very appealing.

#### 5.1 Setting Up the Speech Recognition Software

Speech Recognition software requires that the user undergo “enrollment” and “training” with the software. The enrollment section requires that the speaker read two to three hundred sentences, so that the program can generate user files based on an individual’s personal speaking style, such as accents, and where he or she places pauses. After completing enrollment, the user then enters the training phase for the rest of the life of the software. As a user dictates to the speech recognition program, he or she will make corrections to what the program recognized. These corrections are used to update the user file. The result is that the program will become more accurate with time.

## 5.2 Recording the Audio

Since the speech recognition system is very sensitive of extraneous noise, the audio from the recorded VHS tape is generally not a good enough quality for speech recognition purposes. Instead, a digital recording of the audio must be used. Some computer (such as a laptop) must be set to record the audio of the presentation. To avoid picking up background noise, it is best to attach a microphone directly to the presenter. Although a headset will work best for this purpose, a lapel microphone will still do a good enough job and is less distracting. It should be noted that this is the only part of the system that requires alterations to a presenters standard lecturing style, this is definitely acceptable as evidenced by the fact that most presenters are used to speaking with a microphone, especially to a large audience.

Additionally, the recorded audio may then be used for the playback of the lecture. This higher quality audio will provide a better viewing experience for the distant students. Depending on the sensitivity of the microphone being used, it may be necessary for the presenter to repeat any questions that are asked by the live students in the class; however this is a common practice for most lectures, especially those that are videotaped.

## 5.3 Recognizing the Audio

It is possible to recognize the speech during the actual lecture; however if the time can be spared, it is much better to do this in an offline manner. If the speech recognition is done off-line, then more time can be dedicated to the task,

which will result in a more accurate recognition. It is much simpler to work with several smaller audio files (several seconds to a few minutes each), rather than one large file. Thus by working off-line, the system is able to use the key frame information in order to split the audio file into several smaller files corresponding to each key frame image.

Modern speech recognition systems are not capable of properly parsing grammar such as knowing where to place punctuation. As a way of getting around this, the programs require the user to speak the punctuation while dictating. (For example, a user must say the word PERIOD at the end of a sentence.) This would not be acceptable for a lecture setting. Instead, the system will ignore the grammar of the lecture, and encapsulate each part of the recognized text into a set structure. This way a set of spoken text becomes associated with each key frame, which can be used to extrapolate information about the lecture during that time frame and allow the distant students to search through each set or sets of text for particular key words. The following method is used to extract the text from the audio file and generate the key frame sets:

1. Digitally record the audio lecture
2. Use the timing information from the key frames to split the audio file into corresponding audio segments
3. Individually process each audio segment with the speech recognition program, generating a text file for each audio segment
4. For each file, create a “set” object and look at each word in each text file and perform the following test:
  - (a) Remove any punctuation from the word (i.e., period, comma, etc.)

- (b) Remove any capitalization from the word
- (c) If the size of the word is less than 3 or the word already exists in the set object, discard the word
- (d) else add the word to the set

Since the text that is recovered from the audio does not maintain the original grammatical structure, and the lecture cannot be reconstructed from this audio alone, it is not necessary to store much of the text that was recognized (such as redundancy in words). A set, as a data structure naturally supports this idea. Generally, words of interest will be larger than three letters in length, by removing the smaller words from the set, common words such as “the,” “and,” and “if,” etc. will not be included. By removing punctuation and capitalization, more redundant words are removed from the set structure. The end result is a set of key words or phrases that can be associated with each key frame. Many of the same operations that have been defined from the Optical Character Recognition section of this system can also be defined in the same manner for the speech recognition.

## 5.4 Chapter Summary

This chapter described the speech recognition part of the distance education system. Before a new presenter can use the speech recognition portion, he or she must first enroll with the speech software, reading several sentences to the computer. After that, the user may want to use the software for some amount of time, in order to help increase the accuracy of the recognition. While the presenter is giving a lecture, a digital recording of the lecture must be produced. This recording can be used for the Internet playback of the lecture. The audio

recording is then split into several parts, based on the key frame information, and analyzed separately. The final result is that each audio segment is converted into a set of text that can be associated with each segment of the lecture. These sets can be used to build inferences about the lecture (which will be described in the next chapter) or the student can then perform certain operations, such as searching on these sets. The operations that can be performed are similar to those that can be done with the output of the Optical Character Recognition portion of this system.

## CHAPTER 6

### Lecturelets: Putting it all Together

Up to this point, several separate seemingly disjoint methods for mining information from a video have been presented. In this chapter, the concept of a “lecturelet” is introduced. This concept will concretely tie together all of the methods that were presented, showing how they are all related to each other. The lecturelet will also be used to demonstrate how a distant student can take full advantage of the multimedia aspects of the video lecture, which could not be done with a videotape alone.

#### 6.1 What is a Lecturelet

The abstract definition of a lecturelet is a small, discrete unit of a lecture. Each lecturelet will encapsulate one single idea or thought, as well as a particular state of the lecture during that given time. Each lecturelet will contain various types of information, in order to define the state of the lecture that the lecturelet represents.

While this system is the first one to use the concept of a lecturelet, the definition is intentionally left as an abstract one. In this way, new methods can be introduced for mining information from a lecture video, and those methods

can then be introduced as new pieces of a lecturelet. Additionally, the abstract definition leaves room in the future for an entirely new distance education system to use the concept of a lecturelet and completely redefine the parts.

## 6.2 Getting Lecturelets

The first step to implementing a system with lecturelets is to actually define a “lecturelet trigger,” i.e., when one lecturelet ends and the next one begins. Each individual system would want to define a different lecturelet trigger. Some examples include sound (look for silence in the audio), visual trigger (new information appearing), or simply give the presenter or a post-production operator the control to decide.

The system presented in this thesis uses a visual lecturelet trigger. More specifically the key frame algorithm is the lecturelet trigger. In this way, a new lecturelet is generated every time that new information appears in the video display. For example, this can be each slide in a PowerPoint presentation.

## 6.3 Parts of the Lecturelet

The following is a list of all the different parts of a lecturelet. This section describes how each of these parts are extracted from the mined information in the video. Subsequent sections of this chapter will describe how each of these individual parts can be used to further enhance the interactivity and intelligence of the video presentation versus a simple videotaped lecture.



1. Image representing visual state of the lecture
2. Order of appearance
3. Temporal information, including duration
4. Set of text that was visually displayed (video text)
5. Set of spoken text (audio text)
6. Audio of the lecture

### **6.3.1 Key frame**

The key frame shows the visual state of the lecture during the duration of the lecturelet. In addition to the visual information that the key frame provides, each key frame also has associated with it an ordering and temporal information. The ordering of the key frame corresponds to the ordering of the lecturelet as well. The temporal information gives the lecturelet a specific duration. In addition to helping with playback of the lecture, the information extracted from the key frames gives a means of identifying a lecturelet (either by order or time).

### **6.3.2 Optical Character Recognition**

The optical character recognition that is performed on the key frame allows the lecturelet to “know” what was written. The text can either be displayed as part of the playback of the lecture, or it can be encapsulated in a set in a similar manner to the one described for the speech recognition output. The text can

be used for searching or indexing the lecture. Additionally, the method includes steps for extracting graphics (clip-art) from the image. These graphic files can also be associated with the lecturelet, or some type of graphic analysis algorithm can be used to further process the image and extract even more information.

### **6.3.3 Speech Recognition**

The text output of the speech recognition can be used in almost identical ways to the OCR output. The text can either be displayed or simply treated as a set. As with the optical character recognition, the speech output can also be used for searching and indexing.

### **6.3.4 Additional Components**

As mentioned before, the abstract definition of a lecturelet allows for additional components to be easily added to the lecturelet concept. For example, recent versions of PowerPoint include an option to associate notes with each slide. These notes can be incorporated as another part of a lecturelet. If new methods for extracting information from a videotaped lecture are introduced, the output of those methods can be easily included as well. Figures 6.1, 6.2, and 6.3 show three lecturelets from a lecture about 3 different programming languages.

- **Duration:** 10004
- **Video text:**  
BASIC Simple to use  
Allows anyone to  
progrmm
- **Audio text:**  
Basic is the first  
language that would  
like to talk about it is a  
simple to use  
programming language  
this simplicity gives  
anyone the ability to  
program

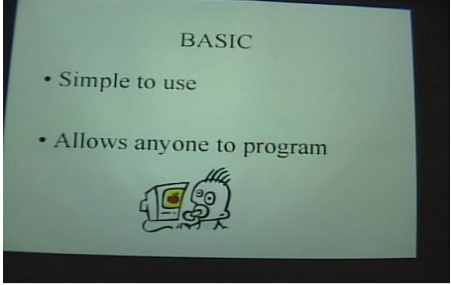


Figure 6.1: Lecturelet containing information about BASIC

## 6.4 Operations on Lecturelets

There are many different operations that can be defined on lecturelets. Up to this point, ideas out of set theory have been alluded to, this is because the lecturelet concept supports many ideas of sets with little or no change.

The first operation to define for the lecturelet concept is union. Whenever a union is performed on all of the lecturelets (assuming that the order of appearance is preserved) then the output is the original lecture itself. This has already been demonstrated in the chapter on Key frames, where it was shown how to reconstruct the entire lecture. An interesting point to note about the union is the hierarchy that naturally occurs. Lecturelets are unioned to form lectures; lectures can be unioned to form a course, courses can be unioned to form a subject or area of study and so on in that fashion.

- Duration: 12228
- Video text:
  - Prolog
  - Myl-
  - Logic mg
  - Closed world assumption
- Audio text:
  - The next language is
  - prologue which means
  - logic programming
  - prologue is based on the
  - closed world assumption
  - which means that
  - anything is not stated as
  - being true is assumed to
  - be false

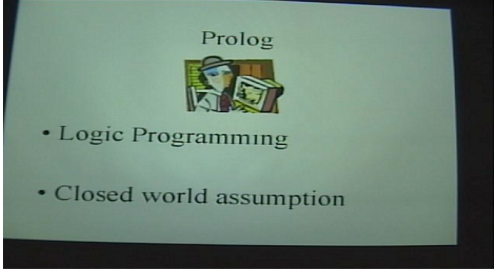


Figure 6.2: Lecturelet containing information about Prolog

Individual parts of the lecturelet can be unioned with other parts, or with other lecturelets. For example, the video and audio text sets can be unioned to form a large search space. Further, those sets can be unioned with the other sets of other lecturelets.

The next major operation is intersection. When two lecturelets are intersected with each other, the common information between the two can be found. This can help similarities between sets of lecturelets and perhaps even relationships. As with the union, the intersection operation can also be defined for individual parts of a lecturelet. Again, looking at the example of the two sets of text (audio and video), the intersection space can help provide greater confidence when a search is performed (searching for words that are both spoken and displayed during a lecturelet).

Subtraction of two lecturelets will yield similar results to the intersection; however this should not be unexpected, since set subtraction is defined in terms

- Duration: 9774
- Videotext:
  - 'g Java
  - Developed by Sun
  - Descended from C and C -++
- Audiotext:
  - . Only there is Java is language was developed by Sun Microsystems and is a direct descendant of C and C++

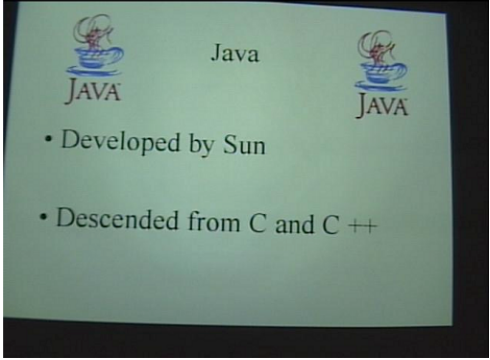


Figure 6.3: Lecturelet containing information about Java

of intersection. If subtraction is defined on the key frame images of the lecturelet, then repeated slides can be found. This usually indicates that the presenter has moved backwards through the presentation. Although a new lecturelet should be formed, the same key frame can be used for both lecturelets, thus further reducing the size of the output presentation.

### 6.5 Implementing Lecturelets and Their Functions

The operations discussed above were implemented with a Java program. The primary purpose in mind for this program was for a student to be able to search through a lecture. Lecturelet was defined as a program class, with each of the components being one of the parts of the object. A lecturelet is able to display information about itself (ordering, duration, text sets, etc). The current

implementation of this program allows lecturelets to perform and output the intersection of the audio and video text sets.

The main program, which uses the lecturelet class, has more functionality than the lecturelet class alone. This program is responsible for loading all of the lecturelets into the computer memory and serve as an interface between the students and the lecturelets. The main program will perform various types of searches (union, intersection, etc.) across all of the lecturelets. Additionally it will perform comparisons (unions and intersections) between lecturelets.

Java was chosen as the programming language to implement the lecturelet concept for several reasons. The first is Java's cross comparability across popular operating systems. It is unreasonable to expect students to all be using the same operating system. As with the choice to use RealPlayer for the playback of lectures, it is desirable to allow the student to use whatever computer system he or she is most comfortable with in order to minimize difficulty with the distance education experience. Along those same lines, Java is also well known for its ability to be ported as web applications, or "applets." Since this system is set up to have the lectures distributed on-line, it only makes sense for as much as possible to be web based.

Next, the lecturelet concept fits very nicely with the object oriented paradigm that Java subscribes to; it is very natural to define an object class "lecturelet," which has all of the components of the abstract definition of this systems lecturelet.

Finally, Java provides implementations of abstract data types that work very nicely as components of a lecturelet. For example, the linkedlist object in Java can be used to implement the text set notions of the lecturelet. Each key phrase from the OCR or speech file for an individual lecturelet can be placed inside a

linkedlist. Since the linkedlist provides a method “LinkedList.existsin(Object)” that will return true if the list contains the object, and false otherwise, this idea can be used to maintain the set nature. This method is also used to implement the searches through the lecturelets.

Figures 6.4, 6.5, and 6.6 show screen shots of the Java program output, after being run on the lecturelets from above. 6.4 is the intersection of each lecturelets text set. In 6.5 a search window has popped up, so that the user can search for text and finally in 6.6 the searched text, “basic,” was found an a new search window appears. The program returns that basic was found in the video and audio text of lecturelet 1.

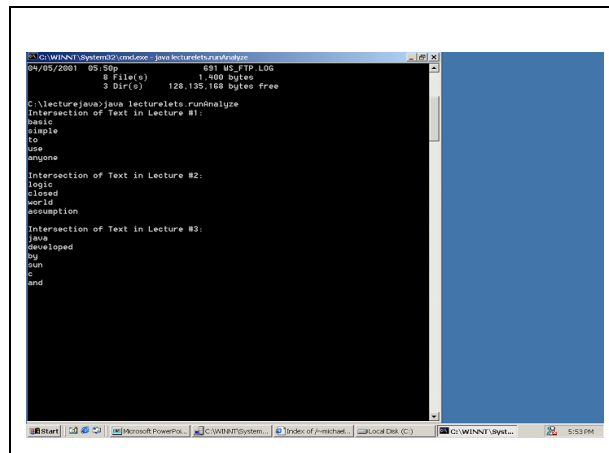


Figure 6.4: Java program showing text set intersection

## 6.6 Chapter Summary

In this chapter, the concept of a lecturelet was introduced. In abstract terms, a lecturelet is a small, discrete unit of lecture. The definition of a lecturelet is intentionally left abstract, so that other education systems can implement ideas

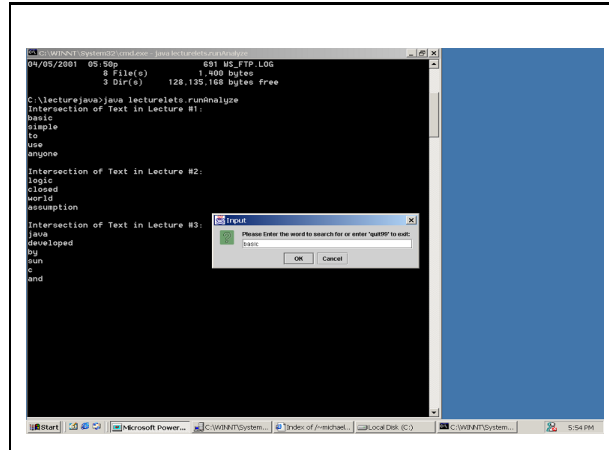


Figure 6.5: Java program showing search window, text to search for is “basic” different from the ones presented in this thesis, and still be able to use this concept.

In order to define the boundaries on lecturelets (when one ends and the next begins), some lecturelet trigger must be defined. The lecturelet trigger simply needs to be some cue that indicates there is new information, that should be encapsulated separately. Triggers can include looking for silence in the audio, new visual information, or simply defined by a human operator. The system presented here uses the key frames algorithm as an implementation of a visual trigger.

The actual components of a lecturelet must be defined by the particular system being designed. This system uses the various methods presented thus far in the thesis to define lecturelet components. They include a single image, ordering, and temporal information which is found from the key frame algorithm; video text and graphics, which are found from optical character recognition on projection systems; and finally audio text, which is found by performing speech recognition on the audio of the lecture.



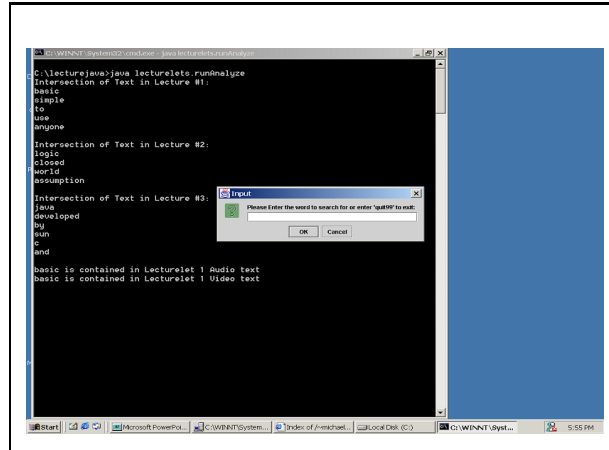


Figure 6.6: Java program showing results of search window, and new window ready for next search

Various operations that are derived from set theory can be performed on the lecturelets, in order to extract information from and make inferences about the lectures. These operations include union, intersection and set subtraction. The union and intersection operations can both be used to define various types of searching. The set subtraction can be used to determine if the presenter has moved backwards in a presentation or has repeated the same slide.

The concept of a lecturelet, as used by this system, as been implemented in as a Java program. This program has an object class “lecturelet” defined, which contains all of the various components of the lecturelet. This class is capable of displaying its information, and searching for words within the two text sets. The main program serves as an interface between the lecturelets and the student, and provides comparisons between lecturelets. Java was chosen as the language to implement this concept in, because of the cross platform compatibility, web capability, object oriented nature, and the abstract data type implementations that Java provides.

## CHAPTER 7

### Conclusions

This thesis presented a new distance education system. The system has most of the advantages of both web based and video based systems, with as very few of their disadvantages. Because of the compression involved, the output of the system can be placed on-line (including video), even under low bandwidth restrictions. Since the lecture is placed on the Internet, there is delay time involved is reduced from days or weeks with a videotape, to minutes (or at worst hours), since the presentation simply needs to be downloaded. The subject content of this thesis will make it interesting to educators, the large portions of the computer science community, and finally students who enroll in distance education classes.

First, a method for extracting key frames was presented. Each key frame represents some change in the visual state of lecture. For example, in the case of a PowerPoint presentation, a key frame would be generated every time the slide changes. The idea is that each time new information is presented, a new key frame appears.

Next the key frames are run through a cleanup process, so that they can be read by an optical character recognition program (OCR). The output of the OCR method is both the text and compressed version of the graphics that have been

presented during the lecture. This output can be used to analyze, index, and search through the lecture.

At the same time as the OCR step is being performed, the audio of the lecture is also run through a speech recognition program. In this way, a text record of all the speech is given. As with the output of the OCR, this text can be used to analyze, index and search through the lecture.

Finally, the concept of a lecturelet was introduced. A lecturelet represents a small discrete unit of lecture. For this system, the key frame algorithm is used as the lecturelet trigger, or function that determines when a lecturelet has ended and a new one has begun. All of the methods presented for data mining in this thesis combined represent the different components of the lecturelet. These components are: a single image, order, and temporal information, all of which come from the key frame algorithm; a set of video text and graphics from the OCR processing, and finally a set of audio text which comes from the speech recognition. Various operations derived from set theory can be performed on the lecturelets and the information that they include. Some of the operations include union and intersection, which enables searching through the lecture. The lecturelet concept was implemented as a Java program. This program defines lecturelet as an object class, with all of the components and capabilities of abstract definition of the lecturelet, as well as a main program which serves as an interface between students and lecturelets. This main program allows students to search and navigate through the video lecture.

The system presented here will have a positive impact on distance education technology, as it combines several positive pieces of other systems, while avoiding the negative aspects. As well, a lecturelet is a powerful concept that adds a new level of intelligence to videotaped lectures. The natural hierarchy of the lecturelet

(being one level below a lecture) fits very well with current education paradigms. A lecturelet works well in computer science as well, as demonstrated by natural use of the lecturelet in both set theory and object oriented sense.

## LIST OF REFERENCES

- [1] Michael N. Wallick; Niels da Vitoria Lobo; Mubarak Shah, "Computer Vision Framework for Analyzing Projections from Video of Lectures." *Proceedings of the ISCA 9th International Confernece for Intellegent Systems*, Louisville, KY, June, 2000.
- [2] Syeda-Mahmood, Tanveer; Srinivasan, S.; Amir, A.; Ponceleon, D.; Blanchard, B.; Petkovic D, "CueVideo: A System for Cross-Modal Search and Browse of Video Databases." *Computer Vision and Pattern Recognition (CVPR)*, Hilton Head Island, South Carolina, June 2000.
- [3] Ju, Shanon X; Black, Michael J.; Minneman, Scott; Kimber, Don, "Analysis of Gesture and Action in Technical Talks for Video Indexing." *Computer Vision and Pattern Recognition (CVPR)*, Puerto Rico, June 1997.
- [4] Victor Wu; R. Manamtha; Riseman, Edward M., "Finding Text in Images" *Proc. of the 2nd ACM International conf. on Digital Libraries (DL'97)*, 1997
- [5] Victor Wu, R. Manmatha. "Document Image Binarization and Clean-up." *Proc. of SPIE/EI'98*, San Jose, CA, Jan. 23-30, 1998.
- [6] Mittendorf, Elke; Schuble Peter; Sheridan Praic, "Applying probabilistic term weighting to OCR text in the case of a large alphabetic library catalogue." *ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995.
- [7] Burhalew, Chris Dr. and Porter, Alan, "The Lecturer's Assistant." *SIGCSE Bulletin*, Volume 26, Number 1. Phoenix, Arizona, March 1994.
- [8] Robler, Toms; Fernndez, David, "Using Multimedia Communication Technologies in Distance Learning." *SIGCSE Bulletin*, Volume 29 Number 23. Uppsala, Sweden, September 1997.
- [9] Murray W. Goldberg and Sasan Salari, "An Update on WebCT (World-Wide-Web Course Tools) - a Tool for the Creation of Sophisticated Web-Based Learning Environments." *Proceedings of NAUWeb '97 - Current Practices in Web-Based Course Development*, Flagstaff, Arizona June 1997.

- [10] Ma, Wei-hsiu; Lee, Yen-Jen; Du, David H.C.; McCah, Mark P., "Video-Based Hypermedia for Education-on-Demand." *IEEE Multimed*a, Volume 5, Issue 1. Jan - March 1998. Pages 72 - 83.
- [11] Siddiqui, K.J.; Zubairi, J.A. "Distance Learning Using Web-based Multimedia Environment." *Proceedings of Working Conference on Academia/Industry Research Challeges 2000* 2000, Pages 325-330.
- [12] Florida Engineering Education Delivery System (F.E.E.D.S.)  
<http://feeds.engr.ucf.edu>