A Computer Vision Framework for Analyzing Projections from Video of Lectures

Michael Wallick, Niels da Vitoria Lobo, Mubarak Shah
School of Computer Science, University of Central Florida, Orlando FL 32816
[michaelw, niels, shah]@cs.ucf.edu

Abstract

The overhead and computer projectors have become an essential element to the classroom and corporate settings. Users of web−based classes and conferences would like to have access to the projector tools. The bandwidth required to transmit these projections is too large to be useful. This paper proposes a framework for processing images displayed by overhead and computer projectors during presentations. In this way, the projectors can be easily rebroadcast over the internet without loss of quality due to compression. This process requires determining the areas of text, binarizing those areas, and performing Optical Character Recognition on the final image. Several examples are shown, including both successful and unsuccessful data sets. A discussion is also included which explains all of the results.

1. Introduction

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 [1]. The ultimate goal of this paper is to be able to take a video of a lecture and create an output of all the extracted text and graphics from the overhead projections after the both have been recognized.

The motivation of our system is the following: the output can be used in many different ways, such as distance learning, video segmenting/indexing, and reconstruction of overheads in videos. When a video is broadcast over the World Wide Web it is compressed so that the video can be viewed streaming, or downloaded quickly. This compression makes it difficult to view the projections in a web broadcast. As a way to address this, if the text and graphics are analyzed separately, then the compressed video can be broadcast with the projection displayed separately. In this way the web user can click on the blurred image to see the expanded view. Similarly when the videotaped lecture is re−played, the projection quality can degrade. Additionally it is time consuming to transcribe everything shown on the overhead. Automating such a process would help greatly. Finally, a video lecture could be indexed based on text shown on the projection. This indexing can also be used to search specific parts of the video. To summarize, the scenario is the following. A lecturer presents using overhead slides or computer presentation. The event is videotaped from some position. The system is intended to analyze the tape and produce an output file of the text.

This paper proposes a framework for extracting the text and graphics from an image of an overhead projector and produce an OCR ready image. This involves edge detection, building connected component regions, and finally segmentation of the display from the background of the image.

2. Related 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 in order 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]. Our 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 [7] or other type of document analysis. The projectors that we have focused on in this work have a unique lighting setup (discussed later in this paper), which can cause problems with the method proposed [7]. Ultimately a system could be developed which uses our method, as well as others to automatically extract information from videotaped lectures.

Distance learning has increased over the past several years. With new multimedia and web based technologies it is becoming easier and easier to offer courses over the Internet [5]. Programs such as WebCT [6] 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 of these factors would contribute to the inability of students to read information from the overhead projector. Again, our system would enhance the distance learning process.

3. Capabilities of Optical Character Recognition Systems

Our system uses Optical Character Recognition (OCR) as one of its components. The algorithms used for OCR are well developed and proven. In fact 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. Images that are not of documents (unconstrained images) will almost always cause the OCR program to fail

Several OCR programs on the market today are able to recognize text by use of Principle Component Analysis or some similar recognition means [4]. 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 read handwriting, our method will properly extract text that is either type or hand written

The overhead projector offers both help and hindrance to situations where computer vision is being used to analyze video lectures. Projectors are generally used with the lights shut off, so that the audience can easily read the projected text and graphics. The darkened room, coupled with the immense amount of light that is given off by the projector makes the area that we are interested in easy to locate in an image. That area simply consists of the brightest pixels in the image. The design of the projector is where the problems are introduced. The overhead projector is 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 the unprocessed image to be unrecognizable by the OCR program. This lighting variance is shown below in figure 1.

Figure 1: Image of overhead projector.

4. Constraints On Data

Although the presented algorithms 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, and generally 300 dpi (dots per inch). The OCR program will not properly recognize low−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.

The algorithms presented makes no assumption about camera position or location. For practical uses, the camera should not be in the field of the projector, as that will block the image. Additionally, the camera should not be skewed, so that the recorded image is undistorted. Although the algorithm presented would be able to extract the text, the OCR program would have difficulty with such an image. Empirically, we found the best position for the camera was on the arm of the overhead projector, as this does not interfere with the projection or block the view of the audience. A position in the same area works well for the computer projectors.

5. Algorithm For Extraction:

In several steps our method is similar to other work [7], as we adapted their system to work robustly on projected images. Here we first present the overall algorithm used in this framework. After that, we present details of each step as well as the differences between our system and other methods. In figure 2, we show the input image of an overhead projection.

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

Figure 2: Original overhead image, with text and graphic

Step 1. Edge Detect 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 viewable. By extracting the edges from the image, we get an outline of all of the text in the image, 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 is common 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 our work, we used a Sobel edge detection algorithm, 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 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 3 shows the edge image, notice that the background variance (in figure 2) is now ignored.

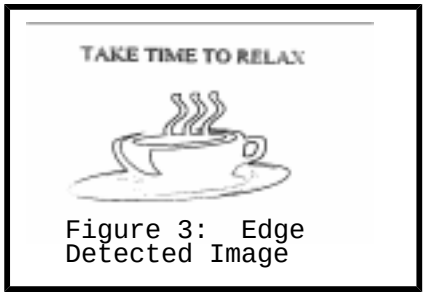


Figure 3: Edge Detected Image

Step 2. Dilate the Edge Image

After extracting the edges from the image, we dilated each edge. This helps in three ways. First, if any of the edges become broken during the previous step, the dilation will connect 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. This is all used in the next step of the process. The figure 4 shows an example of the dilated image.

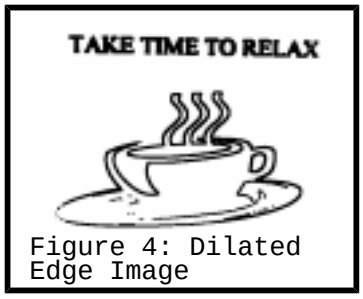


Figure 4: Dilated Edge Image

Step 3. Determine the Connected Components

A simple connected components algorithm is applied to the dilated edge image. This will cause each of the 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.

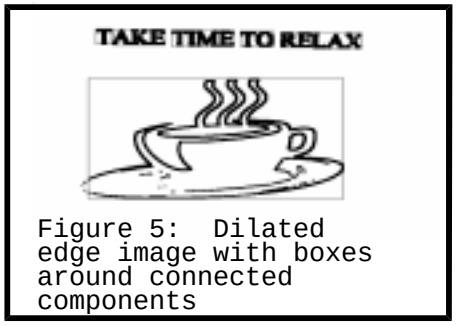


Figure 5: Dilated edge image with boxes around connected components

Step 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. By binary, we mean a two−state frequency (black or white). The rapidly changing light conditions (as discussed earlier in this paper) cause many problems for segmentation. Most segmentation techniques are based on histogram models of the entire image, or a large area [8]. In our image sets, 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 statically based. If the area is too small, the function will not be accurate and again, the segmentation would not work properly.

In order to create a binary image, we propose the following method. 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 lighter than it's neighbors.) A mask size of 20X20 pixels was found to be a reasonable choice for all the images that we tested. While this method is computationally expensive, we have found that is the best way to cope with the problem of varying light.

Below, figure 6 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 it is recovered in the next step. Many other examples are shown in the next section.

Figure 6: Binary Image

Step 5: Determine Text and Graphic Components

After we find the connected components of each region, it is necessary to separate the graphic regions from the text regions. For this step, we employ the aid of the Optical Character Recognition program. The OCR program is used in the following algorithm:

1. Use the OCR program to analyze each of the blocks separately.
2. Check if the blocks 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 and graphics will display different characteristics. Even in the event that a word is misinterpreted by the OCR program, that word will still maintain it's "word" characteristics. Considering these characteristics, we have defined the followings rules for the OCR output of each block. A block that violates any of these rules is marked as a graphic:

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

The justification for each of the three rules are as follows. If there was no information returned (1) then the OCR program was unable to read any part of the block, and returned the same block as a graphics, 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 (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, then we defined an acceptable range of characters that can appear in a projected image (3). 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.

Step 6: OCR the Binary Image

Once the image has been run though 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. The details of the OCR program that we used are discussed in the next section of the paper. Figure 7 shows unformatted text output of the OCR program, we have removed the graphics form 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 section.

Figure 7: Unformatted text output.

7. Results and Additional Figures

For our tests, we used Caere OmniPage Pro (version 9) 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 system can be tested with out without the graphic extraction algorithm (section 5 step 5). All of the images that we tested the graphics extraction on were successful in determining the graphic blocks from the test blocks. Therefore, in this section we only discuss the text extraction.

Since the optical character recognition programs expect 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 test our system's viability, we intentionally used mostly images with a large font size. Therefore, we are break down the results into several subsections. Different criteria for "success" are used for each section. Those criteria are discussed

Group 1: Large Text Image

In the first group, we used large text (50−60−point text). This was to correct for the low resolution of the digital image. We tested this group on approximately 20 image. A success in this group is a character or word that is correctly recognized by our OCR program. All of the text in the output images (step 5) in this group were completely readable by humans. The overall success rate was approximately 95%. Figures 9 and 10 show several examples of pictures from this group.

Group 2: Smaller Text

In the second group, the text is considerably smaller than Group 1 (approximately 36 to 44 point font size). While our system 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. Figure 11 is an example of this type of slide.

Group 3: Poorly Designed Slides

This is the set of images that could not be run though our system. 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 paper.

Because these are ill−suited images, there was no way to classify a successful test. Figure 12 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 than the original image.

Group 4: Problematic OCR Images

While our system makes no assumptions about the type of text, the OCR algorithms are not as forgiving. This type 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. Our framework is able to extract the text from the image, however, the OCR programs are not able to correctly recognize the text. Figure 8 shows a slide that has been skewed.

8. Future Work

With the increased use of the Internet, and on−line" or distance learning classes, this area of research will continue to offer several areas of interesting possibilities. To continue in this field, we would like to expand on the framework outlined in this paper. Ultimately, we would like to see a system that is capable of processing an entire video taped lecture, and preparing it for use in several different ways. By combining other techniques [3], the system would be able to determine the importance and relevance of any given slide, or block of a slide. In an on−line situation, this could be used to automatically create a high light video of what is most important that was covered.

Additionally, once the text is extracted from the video, the information can be used to create a searchable index of the lecture. This way, key phrases, and the time that it was displayed can be quickly found in the lecture.

9. Conclusion

This paper looked at ways in which videos of lectures using overhead and computer generated projections can 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 systems capabilities. 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%).

10. References

[1] Burhalew, Chris Dr. and Porter, Alan, "The Lecturer's Assistant." *SIGCSE Bulletin* Volume 26, Number 1. Phoenix, Arizona, March 1994.

[2] Carswell, Linda. "Teaching Via the Internet: The Impact of the Internet as a communication Medium on Distance Learning Introductory Computer Science Students." *SIGCSE Bulletin* Volume 29 Number 23. Uppsala, Sweden, September 1997.

[3] Ju, Shanon X; Black, Michael J.; et. al. "Analysis of Gesture and Action in Technical Talks for Video Indexing." *Computer Vision and Pattern Recognition (CVPR)*. Puerto Rico, June 1997.

[4] Mittendorf, Elke; Schäuble Peter; Sheridan Páraic, "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.

[5] Robler, Tomás; Fernández, David; et. al. "Using Multimedia Communication Technologies in Distance Learning." *SIGCSE Bulletin* Volume 29 Number 23. Uppsala, Sweden, September 1997.

[6] 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.

[7] Victor Wu; R. Manamtha et. al. "Finding Text in Images" *Proc. of the 2nd ACM International conf. on Digital Libraries (DL'97)*. 1997

[8] Victor Wu, R. Manmatha. "Document Image Binarization and Clean−up." *Proc. of SPIE/EI'98*, San Jose, CA, Jan. 23−30, 1998.

[9] Rohini K. Srihari and Zhongfei Zhang. "Finding Pictures in Context" P*roceedings of the Internaitonal Workshop on Multimedia Information Analysis and Retrieval*, Hong Kong, August 1998.]

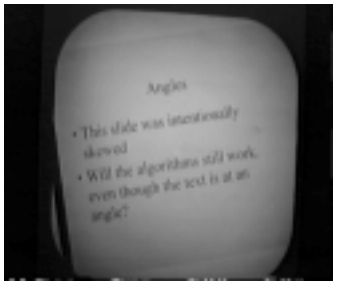
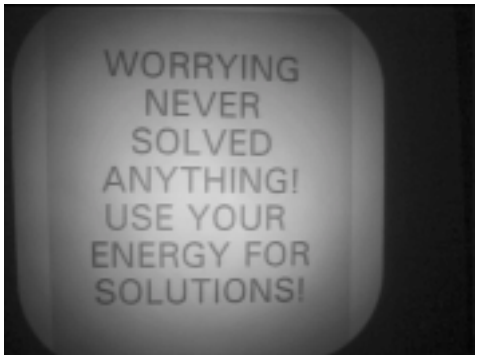


Figure 8: Skewed Slide, not readable by OCR program.



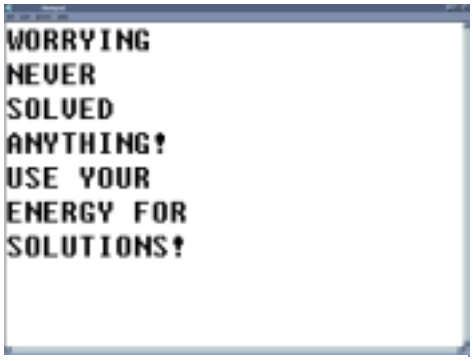
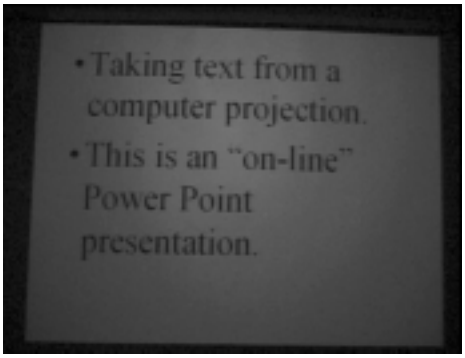


Figure 9: Original Im age and OCR output



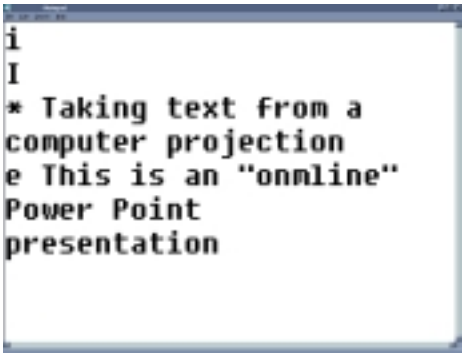
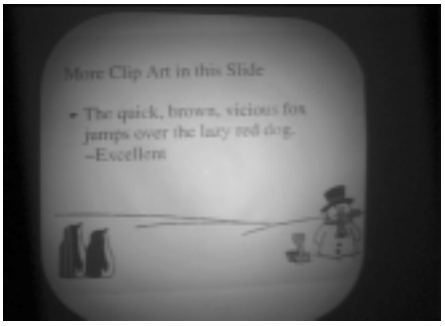


Figure 10: Original image and OCR output



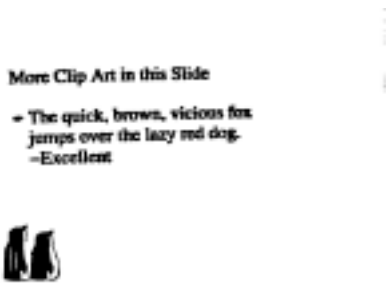
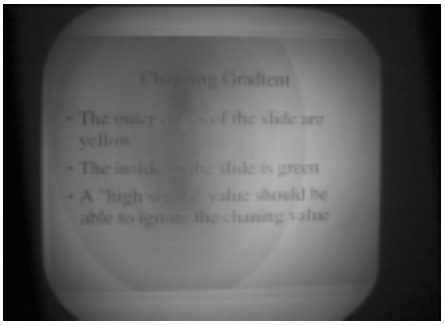


Figure 11: Original Image and Binary Output



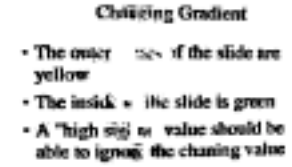


Figure 12: Image with gradient and binary output