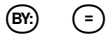
WHO MOVED MY SLIDE? RECOGNIZING ENTITIES IN A LECTURE VIDEO AND ITS APPLICATIONS

by

Qiyam Junn Tung



A Dissertation Submitted to the Faculty of the

DEPARTMENT OF COMPUTER SCIENCE

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2014

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Qiyam Junn Tung
entitled Who Moved My Slide? Recognizing Entities in a Lecture Video and Its Applications
and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Date: 19 December 2014
Alon Efrat

Date: 19 December 2014
Kobus Barnard

Date: 19 December 2014
Chris Gniady

Date: 19 December 2014
Joceline Lega

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.
I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Date: 19 December 2014
Dissertation Director: Alon Efrat

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

SIGNED: Qiyam Junn Tung

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Alon Efrat, for his patience and guidance over the years. He often helped me regain focus when I got too caught up with solving irrelevant details. I would also like to thank Dr. Kobus Barnard for helping me with all the papers and research Alon and I have worked on over the years. His input has been invaluable and has helped direct me towards more interesting research. I am also grateful to the computer vision group for their critiques and library code. I benefited enormously from their collective work and regret I have done little to contribute back. In particular, I'd like to thank Kate Kharitonova for participating in the many hours of discussions we've had and looking over my code and giving me feedback on it. Finally, I would also like to thank my family for being supportive and patient with me and I hope to give back to those who have helped me in some form in the future.

TABLE OF CONTENTS

LIST OF FIGURES 7

LIST OF TABLES 11

ABSTRACT 12

CHAPTER 1 Overview and Background 14
1.1 Overview 14
1.2 Introduction 14
1.3 Identifying Slides 15
1.4 Gestures 16
1.4.1 Laser Gestures 16
1.4.2 Pointing Gestures 17
1.5 Mobile Devices in Education 18
1.6 Energy Conservation in Mobile Devices 19

CHAPTER 2 Prelminaries 22
2.1 Homography 22
2.2 Bounding Boxes 24

CHAPTER 3 Laser Gestures 26
3.1 Single Laser Gesture 26
3.1.1 Identifying and Magnifying Events 28
3.1.2 Speech-based magnification 29
3.1.3 Experiments 31
3.1.4 Conclusion of 3.1 35
3.2 Multiple Laser Gestures 35
3.2.1 System Implementation 37
3.2.2 Study 39
3.2.3 Interview 43
3.2.4 Conclusion of 3.2 44

CHAPTER 4 Pointing Gestures 46
4.1 Pointing Hand Gestures 46
4.1.1 Geometric Preliminaries 49
4.1.2 The algorithm 51

TABLE OF CONTENTS – *Continued*

4.1.3 Identifying the Region of Interest . . . 56
4.1.4 Results . . . 59
4.1.5 Using the Fundamental Matrix . . . 60
4.1.6 Conclusion of 4.1 . . . 61

CHAPTER 5 Further Applications . . . 62
5.1 Energy-savings on Mobile Devices . . . 62
5.1.1 OLED Displays . . . 64
5.1.2 Improving Energy Efficiency . . . 65
5.1.3 Altering the Non-Slide Background . . . 66
5.1.4 Altering the Slide Background . . . 67
5.1.5 Altering and Replacing Slides . . . 68
5.1.6 Methodology . . . 69
5.1.7 Results . . . 70
5.1.8 Conclusion of 5.1 . . . 75
5.1.9 Acknowledgments . . . 75

CHAPTER 6 Future Work . . . 76
6.1 Text-Box-Based Identification of Slides . . . 76

APPENDIX A Appendix A . . . 77
A.1 The Fundamental Matrix . . . 77

REFERENCES . . . 80

LIST OF FIGURES

1.1 Selectively dimming unimportant regions can reduce energy consumption significantly in lecture videos. This figure shows how, once the slide location is known, a video frame (left) can be modified to dim the bright colors of the slide background (right) 21

2.1 The slide as seen as a presentation (left) and its resulting projection on the projector screen (right). Notice that in addition to shifting and stretching, the slide is also slightly rotated due to the camera's viewpoint. 22

2.2 A homography H can be thought of as viewing a planar object from two cameras (the location of the camera centers are denoted by c_1 and c_2). In other words, it gives us some information about the relative camera orientations. Any point p of a plane from the first camera can be mapped with H to a corresponding point p' as seen by the second camera. 23

2.3 The bounding boxes of a particular word, sentence, or image can be found by making the color of the word unique and then taking the image difference. The differing pixels define the location of the word and a bounding box can be constructed from it. 25

3.1 Magnified image 27

3.2 Laser Gesture 31

3.3 The rectangles around the word indicate the bounding boxes (not part of the original slide). The points represent a laser dot sequence moving from left to right. 31

3.4 This figure illustrates how our algorithm works. For the sake of clarity, only a subset of all possible lines are drawn. 32

LIST OF FIGURES – *Continued*

3.5 This figure shows a transcript where the speaker reads off the bullet "Fruits can be eaten fresh." The ground truth alignment, shown in blue, will align the words to the bullet. However, the automated speech recognition system might only detect "eaten" correctly but miss the "fresh" and erroneously interpret "flesh" as "fresh," causing a misalignment. In this scenario, the number of milliseconds where the ground truth and estimate intersect is considered the true positive, the number of milliseconds where there is only blue is a false negative, and the number of milliseconds where there is only red is a false positive. 32
3.6 Laser pointers can be used to get a response from students (top). A teacher can create ad hoc regions (the four boxes lightly drawn with a green marker) for a selection task that will be automatically identified by our system, even in sub-optimal lighting conditions (center); our system will then identify the individual laser pointers and count the number of points in each cluster, giving immediate feedback to the presenter (bottom). 41
3.7 Output of our system tallying answers to a multiple choice question. . 43
3.8 Output of our system tallying answers to geographical question. Our system makes it easy for a teacher to add regions (the light blue boxes) to a slide. 44

4.1 An application of finding the pointing gesture. Once a text bullet is identified, it can be backprojected and magnified into the video to improve legibility. 47
4.2 An example of all 5 gestures. From left to right: pointing with an arm with a bent elbow, a stick, a single finger, a closed palm, and an open palm. 48
4.3 An example of a lecturer using a stick to point to a region of interest in a slide. The white dot in this region is from a laser pointer we have used to groundtruth and indicates where the lecturer is pointing to. . 49
4.4 A lecturer points to the *reference point* p on the screen. This point is also the intersection of the line ℓ_1 containing the pointing stick (or the arm) and the line ℓ_2 containing the shadow of ℓ_1. This fact is used by our algorithm to find the point p' which is the projection of p on the camera image plane. 51
4.5 A lecturer points to the screen with his palm. 52

LIST OF FIGURES – *Continued*

4.6 The same frame as Figure 4.5, after background subtraction and finding shadow and non-shadow pixels. The foreground pixels that are shadows are colored in blue while the rest of the foreground is colored red. 53
4.7 The reference point is found by searching for long line segments and their intersections. The algorithm to find these segments works by finding the locally leftmost points in the foreground (top right) and, for each leftmost point, finding the longest line segment from that point (bottom left). Each resulting segment then extended to a line and is a candidate for a pointing object or its shadow. The intersection points of all pairs of lines each contribute a vote to each bounding box and the box with the most votes is considered the reference point. This figure is an illustration of the main ideas of the algorithm and is not actual output from the program. 55
4.8 After the shadow and the non-shadow segments have been identified, we fit lines to the corresponding segments. Non-shadow lines are colored magenta and shadow lines are colored blue-green. The intersection points, colored green, are used to vote where the lecturer is pointing to within the slide. It is worth noting that in this image, both the shadow from the projector light and the shadow from the room lights (bottom blue segment) were correctly identified, which improves the identification of the reference point. 57
4.9 This figure shows 5 frames processed using our algorithm. These were processed on high resolution images to illustrate how our algorithm works. In the first two frames from the left, the lines are fit to the locally rightmost points, such as the lecturer's fingers. The same algorithm can be used to find the tip of a stick, pointing finger, or the end of a closed palm (the 3 last bottom images). Note that applying linear regression would result in steeper lines because the red foreground's mass is mostly distributed in the lower half of the image. 58
4.10 The lines in the image represent the epipolar lines, which all intersect at the epipole. Put another way, the rays of light are emitted from the projetor (outside the frame). In the context of identifying pointing gestures, it helps in limiting the search of matching long line segments to lie along the epipolar line. The red line shows the epipolar line for the tip of the stick. Note that the tip of the stick lies on the same line as the shadow of the tip of the stick. 60

LIST OF FIGURES – *Continued*

5.1 Layout of pixels of different channels for the (left) Pentile matrix and (right) RGB stripe matrix. 64
5.2 Measurements of the power draw of the three primary colors for all sRGB values 65
5.3 The three kinds of regions in a video frame. 66
5.4 The blue channel is dimmed from the background. 67
5.5 The blue channel is dimmed from the non-bullets regions. 67
5.6 A comparison of the original (right) with the backprojected slide (left). . . 68
5.7 Measurement environment 69
5.8 Average area distributed among a slide, slide text and images, and the background areas. 71
5.9 Energy consumption after dimming less relevant regions. 72
5.10 Images whose slide background is dimmed by 20%, 40%, 60%, and 80% from left to right. 73
5.11 Energy consumption of videos after slide backprojection on the Galaxy S2. 74

A.1 The fundamental matrix describes the epipolar geometry of two views. A point in the 3D scene, X, is seen as X_L by the camera on the left, whose projection center is located at O_L. The fundamental matrix describes where X_L might be seen by the camera on the right. The corresponding point is inherently ambiguous because the point X_L could have come from X_1 or X. Therefore, the corresponding location for the camera on the right is a line rather than a point. The line l_{X_L} is constructed from the epipole e_R, the projection of the left camera on the right camera's image plane. In the absence of the fundamental matrix, X_L could be anywhere on the image as opposed to limited to a line. 78
A.2 Because a shadow is formed by the ray of light from the projector (here, centered at O_P), the image of tip of the finger X_{C_F} and its shadow X_{C_S} must lie on a line collinear with O_C. 79

LIST OF TABLES

3.1 Speech Bullet Alignment Results 33
3.2 Usability Results 35

4.1 Gestures in the video 59

5.1 Smartphone Specifications 64
5.2 Video Statistics 70

ABSTRACT

Lecture videos have proliferated in recent years thanks to the increasing bandwidths of Internet connections and availability of video cameras. Despite the massive volume of videos available, there are very few systems that parse useful information from them. Extracting meaningful data can help with searching and indexing of lecture videos as well as improve understanding and usability for the viewers. While video tags and user preferences are good indicators for relevant videos, it is completely dependent on human-generated data. Furthermore, many lecture videos are technical by nature and sparse video tags are too coarse-grained to relate parts of a video by a specific topic.

While extracting the text from the presentation slide will ameliorate this issue, a lecture video still contains significantly more information than what is just available on the presentation slides. That is, the actions and words of the speaker contribute to a richer and more nuanced understanding of the lecture material. The goal of the Semantically Linked Instructional Content (*SLIC*) project is to relate videos using more specific and relevant features such as slide text and other entities.

In this work, we will present the algorithms used to recognize the entities of the lecture. Specifically, the entities in lecture videos are the laser and pointing hand gestures and the location of the slide and its text and images in the video. Our algorithms work under the assumption that the slide location (homography) is known for each frame and extend the knowledge of the scene. Specifically, gestures inform when and where on a slide notable events occur.

We will also show how recognition of these entities can help with understanding lectures better and energy-savings on mobile devices. We conducted a user study that shows that magnifying text based on laser gestures on a slide helps direct a viewer's attention to the relevant text. We also performed empirical measurements

on real cellphones to confirm that selectively dimming less relevant regions of the video frame would reduce energy consumption significantly.

CHAPTER 1

Overview and Background

1.1 Overview

This dissertation presents a collection of related manuscripts and published papers, which are connected in a logical manner. Each chapter introduces a theme such as a laser gesture, pointing gesture, or their applications. The chapter also has introductory paragraphs listing how the manuscript and papers relate to the thesis. The articles that have been published have been altered slightly to make the dissertation more coherent. Specifically, the related works section from all the papers and manuscripts have been combined together for this chapter.

Furthermore, with the exception of research with multiple laser pointers (Section 3.2), the author was a key contributor in both the development of the ideas and the code of the work listed in this dissertation.

1.2 Introduction

In recent years, distance learning has received a massive increase in visibility. Notable examples include KhanAcademy and massive open online courses (MOOCs) such as Udacity, Coursera, World Science U, edX, and Stanford Online. There are also other institutions that simply provide the materials for the course, such as MIT OpenCourseware and Open Yale Courses, to name a few. The material and support range from just class notes to homework, deadlines, and providing automated grading mechanisms. Nevertheless, in most cases, these educational systems provide lecture videos, which are often accompanied with slides.

The number of lecture videos freely available online now number in the tens of thousands. It is increasingly important to be able to search and relate videos by

topics. While there is much research in the area of general video understanding, the relevant information in a lecture video, such as hand gestures that reference text and images in the slide, are too particular to be handled by a general activity recognition system. Furthermore, a gesture in and of itself is of little significance for a user who may be interested in finding other videos related to the word or phrase that the lecturer is referring to. On the other end of the spectrum, while services like Panopto extract text from presentation slides, it requires special software that the lecturer must prepare beforehand. Furthermore, the software has no concept of the lecture scene and the speaker. Because of the strong link between topics and the presentation slides, our work focuses on understanding the scene and events in the subset of lecture videos that have accompanying presentation slides. As a result, we are particularly interested in lecture videos where the speaker does more than simply "read the slides." There is a wealth of information that can be extracted. Simply pointing at words or sentences is evidence that they are important and that they should be a higher ranked video tag, for example.

1.3 Identifying Slides

But before identifying gestures, some preliminary work needs to be done to find the slide location. Specifically, we would need to know the geometric mapping, known as the homography, which we will expound on in Section 2. Fan et al. (2006) show how to find the slide homography by matching local SIFT (Lowe (2004)) keypoints and using RANSAC (Fischler and Bolles (1981)) to solve for the homography for every frame. In some cases, slides may be identical except for color and Wang and Kankanhalli (2009) show how that can be incorporated into matching. Once the location and identity of the slide is known in each frame, one can segment a video by the slide shown, making it an effective way to browse a presentation on the Semantically Linked Instructional Content (SLIC) webpage. Knowing the homography can also be used to improve the viewing experience by backprojecting the original slide back into the video frame, as shown by Fan et al. and Gigonzac et al. (2008).

Additionally, Cheung et al. (2010) show that knowledge of the homography can also be used to deblur the slide portion of the original video rather than backproject a slide. For our purposes, having accurate homographies allows us to determine where the text and image boxes are within each frame of the video. With this, we can identify when the box is being referenced by a laser pointer or hand gesture as we can convert a coordinate on the frame to a coordinate on the slide.

1.4 Gestures

1.4.1 Laser Gestures

One reason for finding laser gestures is that it is the current region of the lecturer's attention. However, identifying the location of the laser gesture is not the only method of discovering the lecturer's focus. For example, Friedland et al. (2004) show how using an electronic chalkboard can be used to record and transmit strokes for educational use. While this gives the precise location of the lecturer's focus, it requires additional technology and set up. In other words, it adds additional work for the lecturer. Furthermore, it limits the scope to a small subset of lecture videos. One of our objectives is to minimize this requirement and make a system that is compatible with lecture videos of many different settings. A laser pointer is an inexpensive tool that is frequently used. It is therefore useful to have a system that finds laser gestures in a lecture video. Specifically, we implement a system to identify gestures and the text or image they refer to on the slide.

In addition to capturing the lecturer's focus, laser gestures can be used as a form of interaction between the lecturer and the students. Research by Zdravkovska et al. (2010) notes that laser pointers can be an effective component of an audience response system (ARS), and that they in fact require less setup than using clickers would. Their system, however, does not include software that can empirically record the responses. We develop a system that takes advantage of the simplicity of laser pointers while automatically capturing quantifiable data for student responses. In order to achieve this goal, we identify and differentiate the multiple laser pointers.

Work by Oh and Stuerzlinger (2002) and Vogt et al. (2003) uses "blinking codes" to uniquely identify laser pointers. This method requires additional time to set up and may require additional hardware, diminishing the cost-effectiveness and simplicity of using laser pointers. Our system requires no additional hardware beyond a mobile device with a camera; all tracking is done via software. A more involved system called LumiPoint, by Davis and Chen (2002), assumes that points detected on a tiled display can be described as a linear dynamical system. LumiPoint uses a Kalman filter to identify multiple laser pointers, where each stroke's next state can be estimated by its previous position, velocity, and acceleration. A similar system was used to process laser pointer inputs to an interactive entertainment system that enabled audience members to play simple games (Maynes-Aminzade et al. (2002)). Our system is computationally straightforward and, since we are concerned with the pedagogical use of evaluating student understanding, is focused on question-based applications.

1.4.2 Pointing Gestures

In many cases, however, the lecturer is the one that directs the attention of the students. And this can be done with just the use of hand gestures. While there are many studies on identifying hand gestures, we limit this section to pointing gestures as it refers to the text and images on the slide. Kehl and Van Gool (2004) show that pointing gestures can be found by finding the line of sight, which is the line from the eyes to the fingertip of the pointing arm. They use 4 to 6 cameras and use the fact that the points of interest, the head and fingers, are extremal points of the silhouette to solve the 3D locations. Their system allows a user in an immersive environment, such as the CAVE (Cruz-Neira et al. (1993)), to change the direction of a flashlight by pointing in that direction.

Nickel and Stiefelhagen (2003) have a more restricted setup, using only a stereo camera to identify pointing gestures. They identify gestures as well as the target of the pointing gesture. Likewise, they found that creating a line from the head to the hand best indicated the lines in which the object lies, compared to using only the

forearm. When the pointing gestures are given, they are able to get 90% accuracy in identifying one of the eight marked objects. Both Nickel and Stiefelhagen (2003) and Kehl and Van Gool (2004) can operate in real-time; however, both models use cameras from multiple viewpoints, which greatly ease the task of finding the reference point. Furthermore, almost all video lectures are recorded with a single camera, which we assume for our work.

Some have tackled this issue by assuming that hand gestures are close to the regions of interest. For example, Wang et al. (2003) identify hand gestures such as pointing, circling, and underlining. During these events, their system zooms in on a hand gesture to improve focus, which they found to improve legibility. Similarly, we identify laser gestures and use it to magnify the text and images of the slide. However, we are more focused on finding regions of interest in the bounding boxes of text and images, but do so by focusing on *pointing gestures*. In particular, we are concerned with the case of when the pointer, whether it is a hand, stick, or laser pointer, is not necessarily touching the region of interest, but simply pointing to it.

1.5 Mobile Devices in Education

While zooming in on a region may help with legibility, it may still be difficult to read lecture material on the small screen of a mobile device as the video is already small and enlarging a blurry portion of the video will only emphasize the artifacts in the video. This is a significant concern as mobile devices have long been considered as an important educational tool and much effort and development have been put into mobile learning (Attewell and Savill-Smith (2003)). Thornton and Houser did a study to show that students benefit from using mobile devices as a learning tool. They sent e-mail lessons to students' phones to promote learning in regular intervals. At the end of the study, they found that 93% of the students found it a useful teaching method. There has also been success in integrating mobile devices into the classroom in a study conducted by Dyson et al. (2009). Students participated in a lecture by texting responses to activities using their cell phones, giving quick

feedback on the understanding of the class. These studies suggest a trend towards using smartphones for educational purposes.

Our system will help direct attention in a lecture video as well as improve the viewing experience. Specifically, we "magnify" the region by backprojecting an upscaled region of the corresponding high-resolution slide image. This would help deal with the issue of viewing on the small screens of mobile phones. The details of how this is done are explained in Section 3.

1.6 Energy Conservation in Mobile Devices

Even without identifying gestures to magnify, simply knowing where the slide is located can also help with saving energy. Energy conservation has been a challenging focus of mobile research due to the ever-growing demand for more features. In an extensive study, Mahesri and Vardhan (2005) identified that the CPU and display consume the most amount of power in a laptop. The LCD screen dominates during idle times, using up to 29% of the total power draw from the backlight alone. For mobile phones, even finding the optimal schedule for switching an active display to an inactive display can decrease total energy usage by 60% (Falaki et al. (2009)). For this reason, much research has been focused on minimizing the energy consumed by displays.

Flinn and Satyanarayanan (2004) propose a method of reducing the amount of energy consumed without dimming the backlight. Specifically, they use different compression settings, such as reducing the frame rate and decreasing the dimensions of the video. Similarly, Park et al. (2005) show that by degrading video quality by 13%, they were able to achieve savings of 42%, measured on an ARM/Linux-based platform. While those techniques can offer energy savings, they degrade video quality. Degrading video quality is not desirable in our scenario as lecture videos often need to be large and sharp enough for text to be legible.

Another potential method to reduce power consumption is through a lowered refresh rate. While higher refresh rates are becoming increasingly popular in home

electronics, this comes at the price of additional power. Han et al. (2009) developed a method to dynamically determine when the content being displayed is static, and estimated power savings to the effect of 300mW when the percent of time that the display is static is over 70%. This could potentially reduce the energy consumed by lecture videos where the video consists of only a slide, but many have a speaker visible who often makes gestures and other motions. Furthermore, there are slides that have videos and animation embedded in them and so lowering the refresh rate on the slide region may not be desirable.

However, organic LED (OLED) displays can have a significant range of power demand from 0.25W to 2W as compared to the CPU's range of 50mW to 600mW (Kennedy et al. (2011)). Furthermore, the energy consumed is roughly proportional to intensity and color of its pixels, unlike the LCD. Although the maximum luminance of OLED displays rarely occurs in the usage of a phone, the phone usually supplies enough voltage to do so. Chen et al. (2012) show that by dynamically adjusting the voltage after analyzing the display contents can save from 19.1% to 49.1% of the OLED power. Considering that video playback on a Samsung Galaxy S2 uses 35.6% of the phone's total power (Duan et al. (2013)), finding ways to optimize display energy usage is an important task.

In fact, substantial savings can be achieved by dimming parts of the display alone. Iyer et al. (2003) found that reducing the intensity for the background and inactive windows in a desktop environment resulted in energy savings of up to 20%. Similar optimizations have been applied to reduce the intensity of non-active portions of an iPAQ display (Harter et al. (2004)). A user study showed that these changes are unobtrusive and sometimes even preferable. A similar technique of dimming regions that are not the locus of attention in video games can also achieve savings of 11% of the display energy (Wee and Balan (2012)).

In addition, Dong et al. (2009) show that significant energy savings can also be obtained by modifying the color scheme of the graphical user interface. In the same manner, for visualization of data, such as weather data or grouping voxels by color, Chuang et al. (2009) show that one can optimize energy by selecting energy-aware

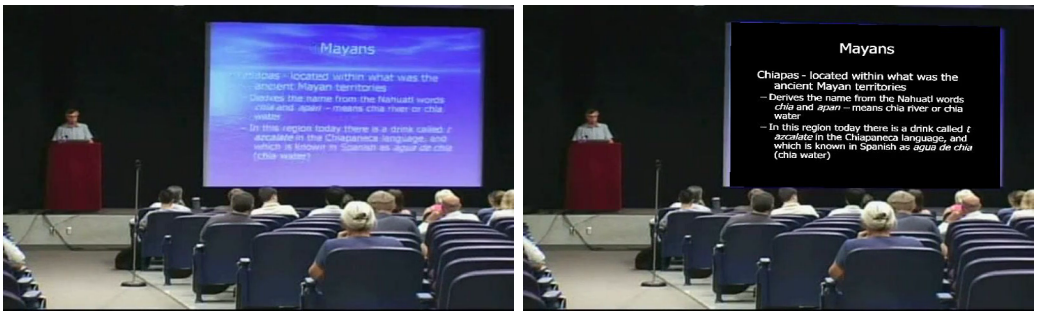


Figure 1.1: Selectively dimming unimportant regions can reduce energy consumption significantly in lecture videos. This figure shows how, once the slide location is known, a video frame (left) can be modified to dim the bright colors of the slide background (right)

but perceptually distinguishable colors. These techniques are not applicable for lecture videos as the color selection was not designed for natural scenes. Nevertheless, selectively dimming colors is the basis of reducing energy for our system, as seen in Figure 1.1.

CHAPTER 2

Prelminaries

Given that magnification and reducing energy consumption depend on locating the slide in a lecture video, we will devote some time in this chapter to explain how to find the slide as well as recognize and separate the slide into meaningful partitions.

2.1 Homography

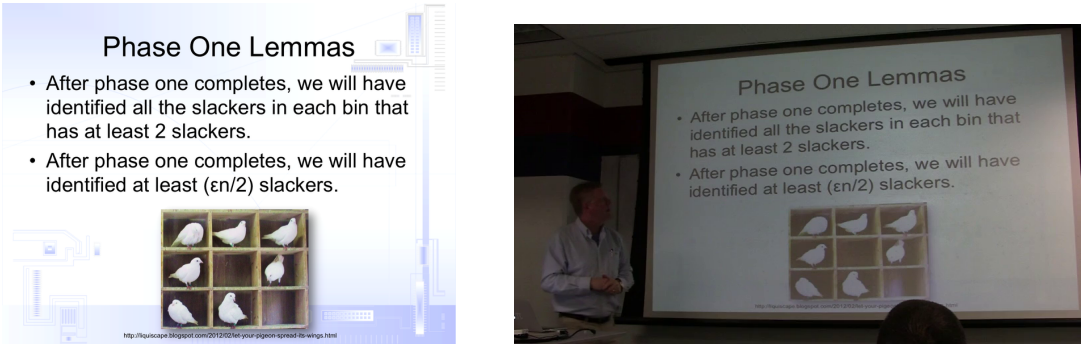


Figure 2.1: The slide as seen as a presentation (left) and its resulting projection on the projector screen (right). Notice that in addition to shifting and stretching, the slide is also slightly rotated due to the camera's viewpoint.

First, locating the slide means to be able to find the transformation of the original rectangular slide image to its resulting quadrilateral in the video frame (Figure 2.1). The mathematical mapping that captures this change is the *homography*, which can be described as the transformation of a planar object from one camera's view to another (Figure 2.2).

Because a change of view, even for a planar object, is not two-dimensional, we will need to change the representation of the image coordinate. When dealing with image points as seen by a camera, it is more natural to use projective geometry as

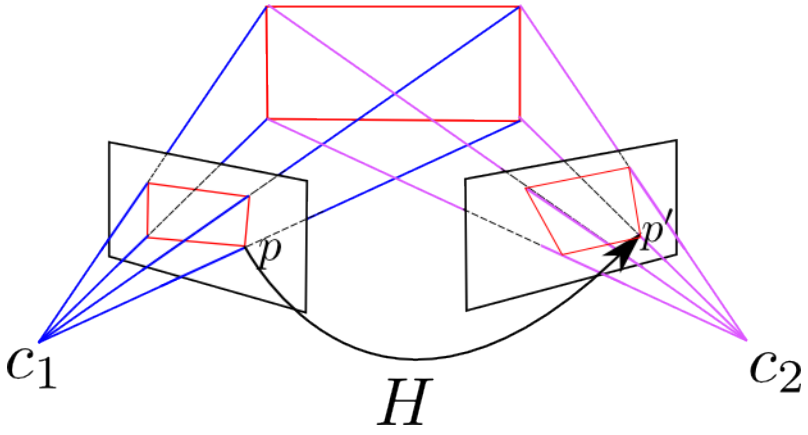


Figure 2.2: A homography H can be thought of as viewing a planar object from two cameras (the location of the camera centers are denoted by c_1 and c_2). In other words, it gives us some information about the relative camera orientations. Any point p of a plane from the first camera can be mapped with H to a corresponding point p' as seen by the second camera.

opposed to Euclidean geometry. Each Cartesian coordinate $(x, y)^\top$ can be converted to its projective equivalent $\lambda(x, y, 1)^\top$, where $\lambda \neq 0$. This representation is known as a homogeneous coordinate. This means that the representation of an image point is not unique as λ can be any non-zero number. Physically, what this has done is to express each image point as a ray, which is constructed from the center of the camera (assumed to be at $(0, 0, 0)^\top$) to the object in 3D. For example, in Figure 2.2, the image point p is represented by the blue line, which is collinear to the camera center c_1 and the bottom right corner of the rectangle in the world.

Since points are represented as rays, we now see why $\lambda \neq 0$, since any point along the ray is a representation of the same object. In order to create an image, there must be an imaging plane. In the case of the homography, where it is a change in viewpoint of the same camera, it suffices to use any arbitrary plane. Therefore, the easiest is to assume that the imaging plane is directly in front of the camera center. That is, the plane is $z = 1$.

To give an example within the context of slide matching, assuming that $(0, 0)^\top$ represents the center of the slide image, then the homogeneous representation of

$(0,0,1)^\top$ also represents the same center. $z = 1$ is the focal length of the camera center, located at $(0,0,0)^\top$, to the center of the image plane.

The knowledge of the image plane is not inherent in the homography, which is a 3×3 matrix, H. Given a point in the first view, $\mathbf{x} = (x, y, 1)^\top$, the point as seen by the second camera can be computed by $H\mathbf{x} = \mathbf{x}'$, where $\mathbf{x}' = (x', y', z')^\top$. Since the camera is the same, the resulting image coordinate can be computed by dividing all coordinates by z', giving $\mathbf{x}' = (x/z, y/z, 1)^\top = (x', y', 1)^\top$. Dividing by z' will not change the representation since it is a homogeneous coordinate and it effectively images the rays at the plane $z = 1$.

It has been shown by Fan et al. that the homography of the slide can be found by using Scale Invariant Feature Transform (SIFT) keypoint descriptors, invented by Lowe (2004). Briefly, SIFT keypoints are local descriptors that are invariant to scale and 2D rotations and robust against slight out-of-plane rotations. Fan *et al.* generate SIFT keypoints for the slide and the frame and find the homography by enforcing the global constraint of the homography among the SIFT keypoint matches. This can be done efficiently using the RANSAC (Random Sample Consensus) algorithm (Fischler and Bolles (1981)), which is a common computer vision algorithm used to fit models that is robust against outliers.

Once the homography is found, the location of every corresponding point on the slide is known. However, the homography does not give us any information about the location of the text and the images. For this, we will need to parse the presentation slides.

2.2 Bounding Boxes

The presentation slide format differs depending on the preference of the presenter. We therefore present a technique that requires minimal reliance on the format of the presentation to extract the bounding boxes of the text and images.

Currently, our method works for Microsoft PowerPoint presentations, although it can be used for any other format. Practically speaking, other formats, such as

Mac Keypoint presentations, can be converted to PowerPoint to find the bounding boxes.

PowerPoint is an XML-based archive and it does not explicitly encode the location of every word or letter as it depends on the particular presentation dimensions, the size of the text box, etc. However, the properties of the text, such as its color, are explicitly encoded by XML. Our observation is that for a given slide, the bounding box of any text t can be found by creating two nearly identical slides. The only difference is the color of t, which is assigned a different color for each slide. When the slides are converted to images, each corresponding pixel is compared and pixels that differ mark the locations of t. A bounding box is then be constructed for this set of pixels (Figure 2.3).

Figure 2.3: The bounding boxes of a particular word, sentence, or image can be found by making the color of the word unique and then taking the image difference. The differing pixels define the location of the word and a bounding box can be constructed from it.

The process for finding the bounding box of an image is similar. In a PowerPoint presentation, images are stored in a separate directory. Unlike text, where each word would need to be parsed and given the appropriate color in the XML format, images are much simpler. Once all the image filenames have been identified in a slide, each image filename can be replaced with a monochrome image. We can therefore repeat this process for images by creating two slides for each image, which is again given a different color.

Because we have the homography, once the bounding boxes are found, we can also identify where each text and image is located within the video frame. With this information, any pointing gesture on the slide can be localized to the word, sentence, or image on the slide.

CHAPTER 3

Laser Gestures

One of the practical applications of identifying slides in a video is being able to partition the video based on the slide being shown. Furthermore, a text search can be localized to a slide. A more fine-grained segmentation can be achieved if gestures are identified. That is, within a video segment of a single slide, a search on a word or phrase can be localized to the point in time where the lecturer is pointing to it.

However, hand gesture recognition is a difficult problem even in constrained settings. As a first step, we develop an algorithm to identify laser gestures, which are easier to identify.

The work for laser gestures (Section 3.1) was published in the ACM Multimedia Conference of 2011 (Tung et al. (2011)).

The research on multiple laser gestures (Section 3.2) was led by Dr. Alon Efrat and Dr. Angus Forbes and the code was written by Saurabh Maniktala and Anisha Goel. The author contributed by designing and running the user studies as well as writing a signficant portion of the manuscript.

3.1 Single Laser Gesture

We present a system that assists users in viewing videos of lectures on small screen devices, such as cell phones. As the participant views the lecture, the system magnifies the appropriate semantic unit while it is the focus of the discussion. The system makes this decision based on cues from laser pointer gestures and spoken words that are read off the slide. It then magnifies the semantic element using the slide image and the homography between the slide image and the video frame. Experiments suggest that the semantic units of laser-based events identified by our algorithm closely match those identified by humans. In the case of identifying bul-

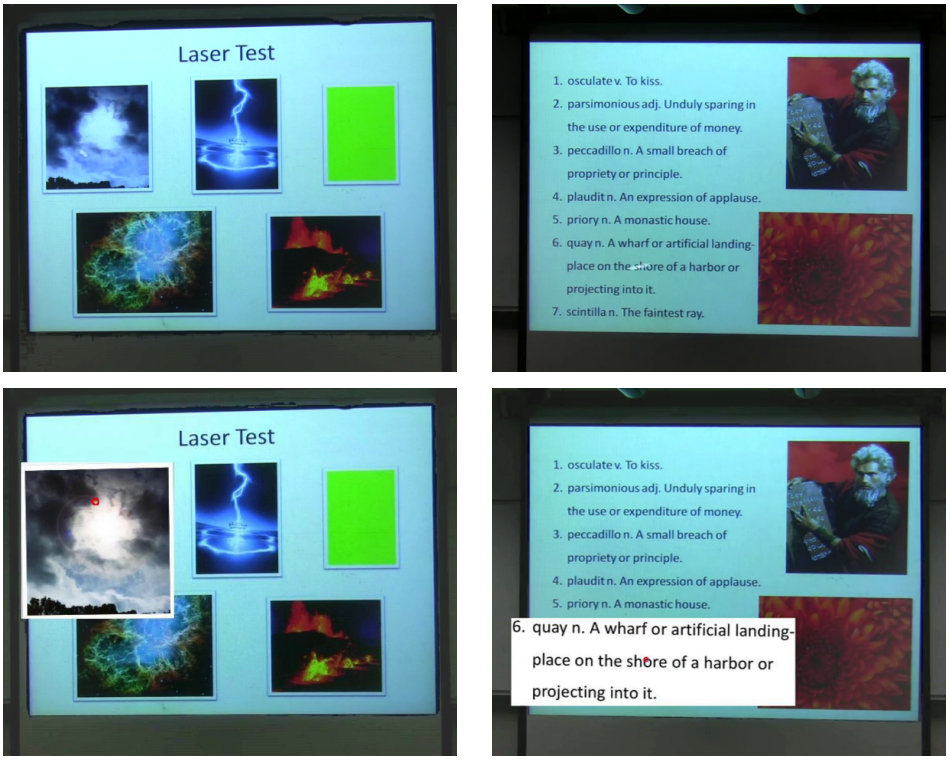


Figure 3.1: Two snapshots from videos played with (bottom) and without magnification (top). A semantic unit (image, bullet, or word) is magnified when triggered by a laser gesture. A semantic unit is also magnified for a length of time when a word from the bullet is read.

lets through spoken words, results are more limited but are a good starting point for more complex methods.

Specifically, our contributions are as follows:

- Identifying **semantic units** in each slide, such as bullet points, groups of bullets, and images.
- A method for robustly identifying the position of each semantic unit on a presentation slide.
- Identifying events, which are a lecturer's attempt to draw attention to a semantic unit, based on analysis of speech transcript and aligning them to times in the video corresponding to each semantic unit. We consider a spoken word *aligned* to a bullet if we know what bullet it belongs to.
- Identifying events based on laser pointer gestures.
- Augmenting the video by backprojecting an enlarged sharp image of this semantic unit, taken from the full-resolution slide.

We have demonstrated the usefulness of our technique towards increasing readability of lecture videos by exposing two randomly selected groups of students to two videos, one with magnification and one without. The assumption is that when the lecturer knows that the laser pointer would be used to enhance the video, he or she would use it in a manner similar to the way it is used in the aforementioned videos. We found that the users tended to remember the text in a slide that was highlighted by a laser pointer better when they watched a video with magnification. We have also tested our algorithm that detects when semantic units are highlighted by a laser pointer and found that the identified semantics units were very similar to those identified by humans. These experiments are detailed in Section 3.1.3.

3.1.1 Identifying and Magnifying Events

Next, we describe how to identify at which frames a semantic unit is being discussed in the lecture video. We achieve this based on speech and laser pointer use. Once identified, we magnify the element in the video frame coordinate system. We use the algorithm described by Fan et al. to find the homographies for slides to frames and thus the time at which the slide is shown in the video. In other words, this

operation describes the relationship between a slide and its projection in the video. The bounding boxes in the slide combined with the homography specify where the semantic units are located within the video frame coordinate system.

In the following sections, we will describe how we determine when a text or image is being referred to.

3.1.2 Speech-based magnification

Here, we discuss how to identify text based on automatic speech recognition systems.

Speech events

In a lecture, the speaker may utter words that appear in a bullet of the slide. When the words in a bullet are read from a slide and are correctly mapped to their corresponding spoken words, we obtain times (i.e. video frame number) for when a bullet should be magnified.

Swaminathan et al. (2010) show how to align speech transcript to the words on slides in the context of improving the transcript. This was done by creating a hidden Markov Model (HMM) for expected phoneme sequences for each slide, allowing for words to be skipped or additional words to be inserted, which often occurs as speakers embellish their main points. This benefits us because once we know which bullet a spoken word belongs to, it also informs us the time at which a bullet is being discussed. At its current implementation, the model only allows for a bullet to be spoken once.

From our experiments, we use the timing information accompanying the speech transcript and follow the algorithm outlined by Swaminathan *et al.* to create time boundaries for each bullet. The boundary for each bullet is created by using the minimum and maximum timestamps of all words in a bullet.

Laser pointer events

In addition to identifying speech-based events (see Section 3.1.2), we also identify events where the laser pointer is used to highlight bullets or images. We use the technique developed by Winslow et al. (2009) to find the laser points. Briefly, it tracks a laser pointer by first computing the difference between the current frame with the average of the last ten previous frames, the idea being that a laser pointer is in constant motion. Then, it looks for the 3×3 block that has the brightest median intensity in the image. To deal with shakiness, the sequence of the brightest points found in each image is then fit to a cubic to remove the shakiness that comes from the speaker's hand. The homography is then used to map the laser point to find where it appears in the slide's coordinate system.

Next, we try to identify which box the laser pointer gesture is highlighting. One simple algorithm is to select the box that gets the most votes. A vote is given to a box if a laser point in a frame is within one of the boxes. However, this algorithm will fail most of the time because of laser gestures can vary significantly in movement. For example, some lecturers may use a laser pointer to underline a bounding box while others may simply wiggle their laser pointer near the box of interest. They might also circle around the box without ever entering the box or may only intersect with a portion of it (Figure 3.2). Clearly, the last case would make a simple voting scheme based on whether a laser pointer falls inside a box impractical.

In other words, the algorithm to select a box must take into account the movement of the laser gesture. Hence, our algorithm will use a voting scheme based on line intersections to determine the semantic unit being discussed. Given a laser *gesture*, a small time interval of continuous laser points (Figure 3.3), we compute a set of line segments from all the pairs of laser points (see Figure 3.4). This provides a notion of the movement of the laser points in the video. For example, the first two points (leftmost points in the figure) fall outside of the box. However, the segment between the two points intersects the box or lies inside the box, so the algorithm counts that as a vote. Furthermore, this method approximates the area of a con-

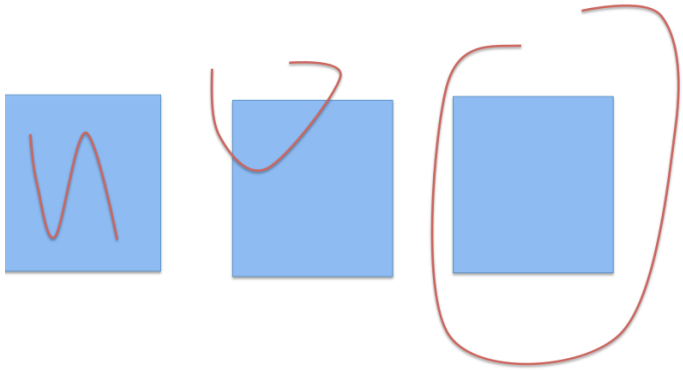


Figure 3.2: The curves represent examples of the paths of a laser gesture. The laser gestures can be arbitrary and do not necessarily fall inside the box. For all three cases, our algorithm can still identify which bounding box the laser is highlighting.

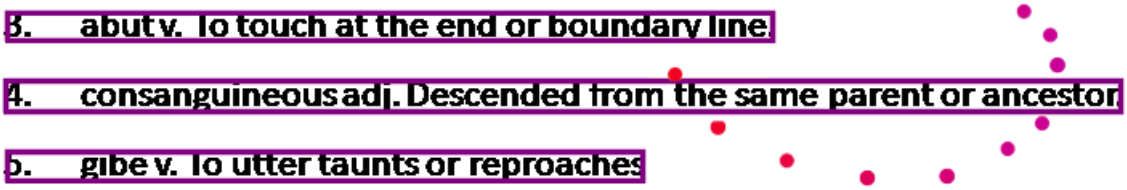


Figure 3.3: The rectangles around the word indicate the bounding boxes (not part of the original slide). The points represent a laser dot sequence moving from left to right.

vex polygon that contains all the laser points. The output of this method can be thought of as a measure of the amount of intersection between two polygons. In the rightmost example in Figure 3.2, the laser points determine a path that curves around a semantic unit's bounding box. Even though the points never fall within the box, it will still get votes from the resulting line segment intersections.

3.1.3 Experiments

We ran three sets of experiments measuring how well the spoken words are aligned to text bullets, how accurately our algorithm identifies semantic units through laser gestures, and how effective magnification is in learning.

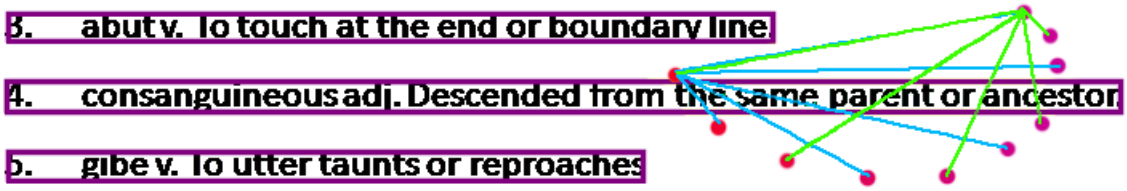


Figure 3.4: This figure illustrates how our algorithm works. For the sake of clarity, only a subset of all possible lines are drawn.

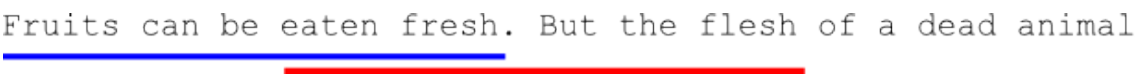


Figure 3.5: This figure shows a transcript where the speaker reads off the bullet "Fruits can be eaten fresh." The ground truth alignment, shown in blue, will align the words to the bullet. However, the automated speech recognition system might only detect "eaten" correctly but miss the "fresh" and erroneously interpret "flesh" as "fresh," causing a misalignment. In this scenario, the number of milliseconds where the ground truth and estimate intersect is considered the true positive, the number of milliseconds where there is only blue is a false negative, and the number of milliseconds where there is only red is a false positive.

Speech alignment

In this experiment, we tested how well our method can identify the time boundaries of a bullet (or part of a bullet) when it is read off the slide. Since our algorithm can only identify bullets whose slide words appear in the transcript, we ran our algorithm on 3 videos in which the lecturer often read off the words of a slide.

We compared the estimated bullet time boundaries to the boundaries created manually, which was done by identifying which bullet words were read from the slide. We use precision and recall to estimate how good the results are. Specifically, precision is $t_p/(t_p + f_p)$ and recall is $t_p/(t_p + f_n)$, where t_p, f_p, and f_n are true positives, false positives, and false negatives, respectively.

For each bullet is an associated estimated time segment and the ground truth time segment. The number of true positives is calculated by the number of mil-

	Precision	Recall
Talk 1	0.290	0.375
Talk 2	0.369	0.409
Talk 3	0.317	0.343
Average	0.320	0.376

Table 3.1: The precision and recall of bullet time alignment.

liseconds The number of true positives is obtained by intersecting ground truth and estimated time boundaries and the remaining milliseconds of the estimated bullet time boundary are considered false positives (Figure 3.5). Similarly, false negatives are the number of milliseconds of a non-bullet time interval incorrectly predicted by our algorithm.

Table 3.1 shows that the average precision and recall are 32% and 37.6%, respectively. While the results may not be good enough to make magnifying semantic units based on speech practical by itself, it provides useful information for more complex methods, such as identifying bullets through topic similarity.

Laser pointer event test

In this experiment, we tested our algorithm's accuracy of identifying semantic units with laser pointers.

Setting. We used 9 short 30-second videos. In the video, the lecturer used a laser pointer to highlight semantic units with simple gestures like circling and underlining. Three graduate students watched and created a ground truth sequence of semantic units highlighted for each video. The sequence of events from each student was in perfect agreement.

Results. To test the accuracy of our algorithm, we checked to see whether it correctly identified the order of semantic units highlighted by lecturer. To do this, we computed the edit distance (using the Unix *diff* program) between the ground truth and the machine generated sequence. Specifically, it counts the minimum number of

edits (insertion, deletion, or replacement) needed to modify the machine generated sequence to the ground truth. The error rate is defined as $error = e/l$, where e is the number of edits and l is the length of sequence of semantic units. There were a total of 8 edits out of a sequence of length 59, which gives us a error rate of 13.6%. However, the errors were due to the fact that our laser tracking algorithm lost track of the laser point for a few frames, changing a single gesture into two gestures. Otherwise, our algorithm yields the same semantic unit sequence as the ground truth data.

Effectiveness of magnification

To measure the effectiveness of the enlargement, we created a questionnaire by sampling GRE-level nouns. We showed each participant a 50-second video with two slides containing around 10 definitions of these uncommon nouns (e.g., "gynecocracy"). To focus the participant's attention to particular nouns, a lecturer would use a laser pointer to highlight them. The screen size were chosen so as to simulate a typical smartphone. Students randomly viewed either the original video or a video in which enlargement was performed on the highlighted bullets. Once the video ended, they were automatically redirected to a questionnaire on the definitions of the highlighted nouns.

To measure the correctness of each group, we counted the percentage of total correct answers, $s = \frac{\#correct}{\#total}$. We labeled "I don't know" responses as neither right nor wrong, so we gave it a score of zero. In our experiments, there were a total of 40 responses. 23 of those saw the original video and 17 saw the magnified video.

From Table 3.2, we see that participants who viewed the magnified video answered more questions correctly and made fewer mistakes. This is reflected by the scores of the users who did and did not watch the magnified video, which are 72.3% and 46.0%, respectively. Assuming that the answers from each group are normally distributed, we use Welch's t-test to show that the scores are statistically significant. The p-value of 0.0092 confirms that participants generally perform much better at remembering the definitions of bullets when they were magnified.

	No Magnification	Magnification
Total Correct	74	86
Total Incorrect	87	33
Score	0.460	0.723

Table 3.2: The table lists the data from the user study. It is partitioned into the group that watched the video with magnification and the group that did not. We allowed students to respond with "I don't know" to prevent excessive incorrect answers and therefore the columns do not necessarily add up to the same sum.

3.1.4 Conclusion of 3.1

We demonstrated a method for magnifying relevant semantic units. Our experiments suggest that our method of identifying semantic units by laser pointer is accurate insofar as human judgment is concerned and when the laser points themselves have not been missed. Though our speech-based events are less accurate, it is a good first step to relating spoken words to bullets. Finally, our user study shows that our method of enlarging semantic units can potentially help users remember the lecture contents.

3.2 Multiple Laser Gestures

An additional use of laser gestures is in facilitating class interaction. In this section, we introduce a low-cost system to enable a range of ad hoc selection and highlighting tasks using many laser pointers in a classroom environment. Our system is able to indicate clusters, tally votes, or highlight selected regions in real-time. It requires only a smart phone or tablet with an integrated camera and is therefore easy to set up. A preliminary study involving a series of experiments conducted in a classroom with 1 teacher and 23 students shows that the system accurately detects user choice with very little set-up time and only a bare minimum of training. Moreover, a survey of the participants of the study finds that the system is both engaging and easy to

use.

Research into the use of audience response systems (ARSs) in classroom environments has shown that, despite some challenges, in many situations it is an effective way to provide feedback to the instructor and to improve attentiveness, increase knowledge retention, and encourage participation amongst students (Banks (2006); Fies and Marshall (2006); Graham et al. (2007); Kay and LeSage (2009); Stowell and Nelson (2007)). A survey regarding the most commonly-used type of ARS, the clicker, overwhelmingly found that teachers and students found it a useful technology to encourage active thinking (Caldwell (2007)).

Clickers, however, can be prohibitively expensive for certain communities (Boatright-Horowitz (2009); DeBourgh (2008)). Recent research has explored the introduction of low-cost technologies for enabling increased participation and engagement in classroom activities, especially in environments that do not have access to more expensive ARS technologies. For instance, Zualkernan (2011) presents an open-source, low-cost clicker-like system called Info Coral; Lin et al. (2011) explores using a Wii-remote as a low-cost device for communicating with an interactive whiteboard; Kreitmayer et al. (2013) introduces a lightweight, multi-device system called a UniPad for interacting with a wall display; Alvarez et al. (2011) and Mundy et al. (2010) make use of mobile technologies for collaborative classroom learning activities; and Cross et al. (2012) describes an implementation using QR-like codes printed on a large card that are then processed with a computer vision system to automatically tally student choices.

Laser pointers similarly have the potential to be used in a variety of different situations where accessibility to technology can present a barrier to participation (Zdravkovska et al. (2010)). The price of a simple laser pointer, including batteries, can be less than $1US, significantly less than a clicker ($15-$35 US). Moreover, laser pointers enable interactions that are not possible through clickers, which generally rely solely on multiple choice selection. And, as shown in research by Chang (2013) (discussing a system involving multiple mice for collaborative interaction), allowing students to view each other's interactions can be harnessed to promote learning

through engagement and competition.

In this section, we introduce an interactive, real-time system that allows a teacher to design an ad hoc selection or highlighting task to gather instant feedback. The system runs on a mobile device with an integrated camera to detect the movement of laser pointers on either a whiteboard or on projected slides, and can, within a few frames, find clusters, tally votes, and indicate selected regions. While we have not established a fixed upper bound on the number of students that can use our system simultaneously, it is effective with at least 23 students with very minimal training for a range of tasks.

We make the following contributions:

1. The idea that humans can quickly and easily find their own "dot" amongst a large number of other similar dots controlled by other students with laser pointers.

2. A system that can be integrated easily, cheaply, and effectively into a classroom situation.

3. A system that enables a range of collaborative activities that involve selection or highlighting tasks.

3.2.1 System Implementation

Our main implentation runs on a mobile device with an integrated camera and is divided into two main phases. The first phase detects the laser pointers by analyzing frames from the real-time video capture and generates a set of coordinates for each laser pointer in each frame. The second phase then processes these coordinates, finding clusters of points, determining whether or not the points are within particular boundaries, and tallies votes. With a touch of a button on the screen of the device, a user (e.g. a teacher) can initiate processing on either the live feed of the camera or on stored videos of an earlier recordings of a task involving laser pointers. In all cases, the camera is expected to stay in place during the detection phase.

In the first phase, our algorithm detects multiple laser pointers using the following steps:

1. Take a sliding window of the past 72 frames (excluding the current frame).
2. For each current frame, take 6 samples from the sliding window. We consider the 72nd, 64th, 32nd, 16th, 8th and 4th frame from the current frame.
3. Convert the frames to grayscale.
4. Add the values of these 6 samples in a weighted fashion, giving more weight to older samples to generate a background frame. (The values of these weights should sum up to one.)
5. Use background subtraction under a threshold tp to generate a foreground of laser blobs.
6. Apply contour detection on blob frame to generate a list of contours and then use the contours to generate a list of approximate bounding polygons for each of the contours.
7. Finally, apply bounding box detection onto the list of approximate polygons to yield a list of bounding boxes each possessing two coordinates, i.e. the top-left and bottom-right.

The list of bounding boxes generated in the last step constitutes the input to the second phase. This set of bounding boxes, indicating the location and size of laser points, should not be confused with user-drawn boxes used for tallying in the second phase. The threshold parameter tp is a user-controllable parameter that is by default set to two-sevenths of the average brightness of current frame. We generate a differential frame in step 5 in order to prevent bright, stationary light sources from being falsely detected as laser pointers.

In the second phase, our system then detects clusters of laser pointers either freely, without consideration of the background slide or whiteboard, or, in "box-mode," in relation to user-drawn boxes (on a white board) or projected boxes (on

a slide). At any time, a user can interact with the system to find clusters and to count the number of points within each cluster. A small bounding box surrounding each point in that cluster is drawn with a distinct color that is associated with each cluster. A parameter k defines the number of clusters (via a real-time k-means clustering algorithm) that the system expects. In cases where no student will have selected a particular choice, the user can update this parameter interactively in order to ensure an accurate counting of selections.

In "box-mode," our system attempts to automatically identify boxes projected on a slide or drawn by the teacher. The teacher can also use our interface to define each box by clicking on a point to define the top-left corner and then dragging and releasing to define the lower-right corner. Although we still detect and visually identify clusters, in box-mode the tallying is done with a simple check that determines whether a point is within a particular box.

3.2.2 Study

We performed a series of preliminary experiments to test our system in a real-world environment, consisting of 1 instructor and 23 undergraduate students. These studies were pedagogically motivated by the desire to have a straightforward, easily measurable way to evaluate students responses to a variety of different types of questions. Teachers often use either pre-made slides or use a whiteboard to present information in an ad hoc manner. We explored student interaction via both of these modalities. In each experiment we asked students to use their laser pointer to indicate their response. We also instructed the students to write down their answers in order to verify the accuracy of our system. Finally we asked the students to fill out a survey with their qualitative assessment of using a laser pointer for each of the tasks. We also asked the instructor to describe his experience using the system.

Experiment 1

Our first experiment measures the time it takes for a student to determine which laser pointer he or she controls. We created a simple study to mimic the visual confusion that occurs when a number of moving colored dots appear. Specifically, students were asked to locate and point to the first letter of their last name and to hold their laser pointer steady once they selected it. The letters were randomly placed so that students would be first required to scan the slide for the correct letters. While humans are not very good at tracking multiple objects (Pylyshyn and Storm (1988); Ware (2012)), for these tasks, students only need to be able to identify their own dot. That is, the potential cognitive difficulty is to locate and track an object while ignoring other moving dots.

Students were almost immediately able to identify their own dot (within approximately 3.5 seconds!) the very first time they tried the laser pointer in our study. Although we did not set out to investigate *how* this happens so quickly, it seems reasonable that humans are able to accurately correlate movements between the hand and the dot emitted from the laser pointer that they are holding. Student responses further supports this idea as some mentioned how moving or wiggling their pointer helped them to identify the laser pointer. Some students also indicated that it seemed somewhat more difficult to do this in other tasks when a larger number of dots were grouped close together. In a future study we would like to test the limits of this ability to recognize one's own dot in regards to: the number of total dots, the speed of the dots, and how close together the dots are. However, these potential limits were not an issue for a classroom with 23 students.

Although we did not separate the pointing and circling categories, it should be noted that the students felt that it took longer to identify their pointer when making circling gestures compared to pointing gestures.

Nevertheless, the results indicate that humans are, on the whole, quite capable of quickly identifying and tracking their own laser pointers.

Experiment 2

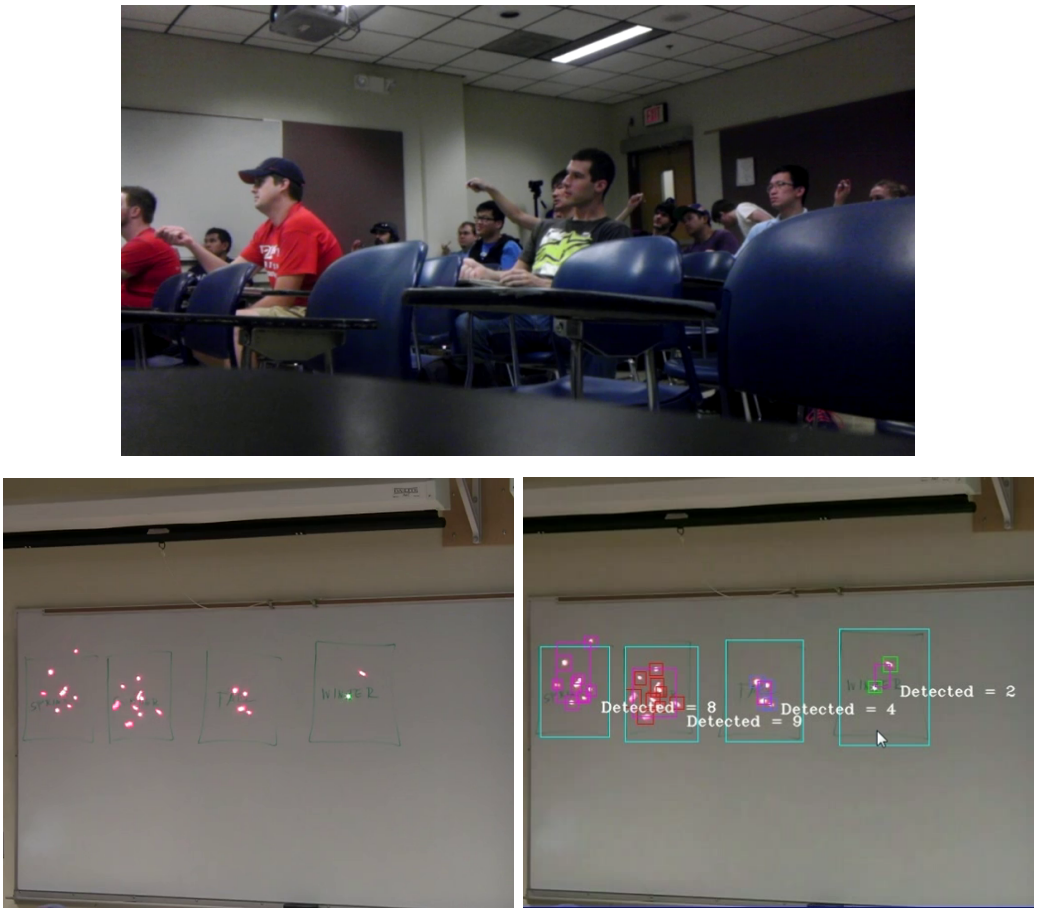


Figure 3.6: Laser pointers can be used to get a response from students (top). A teacher can create ad hoc regions (the four boxes lightly drawn with a green marker) for a selection task that will be automatically identified by our system, even in sub-optimal lighting conditions (center); our system will then identify the individual laser pointers and count the number of points in each cluster, giving immediate feedback to the presenter (bottom).

In this experiment we asked students to vote on a series of choices indicated by boxes that were either drawn in marker by the teacher on a whiteboard or projected on a slide. We also asked the students to record their choices on paper so that we had a baseline with which to compare with the accuracy of our system.

We conducted three tests, each having an increasing number of choices (2, 4, and 12). The first asked them to indicate if they were born on an even-numbered day or an odd-numbered day. The second, which season (Spring, Summer, Fall, or Winter) they were born in. And the third, which month were they born in. Accuracy is computed as the number of correct votes over the total number of votes. An incorrect vote, computed for each box, is the distance between the number of votes in the ground truth and the number of votes output by our system. The accuracy for each test is done by calculating the accuracy for each individual frame and then calculating calculating the mean. We extracted samples at four different times from the video feed. The average accuracies were 91.3%, 89.3%, and 92.8% for the birthday parity, birthday season, and birthday month tests, respectively. The highest accuracies were 94.7%, 95.7%, and 95.7%. Once again, all the students were easily able to recognize their own dot. Some of the inaccuracy was due to the normal fluctuations of motion of the person holding the laser pointer, which sometimes caused the dot to move in and out of the correct position momentarily. On the other hand, surprisingly, a few students were able to keep their dot completely still, in exactly the same place, for the full 72 frames. In this case, this meant that our system would not detect it moving at all, and thus incorrectly categorize them as a fixed light source (and not a dot emitted by a laser pointer). In practice, however, we believe that these numbers could usually approach 100%. Because the system is interactive, a teacher can easily give a countdown to make sure that the students are pointing at the same time. Moreover, if there are obvious mistakes due to misunderstanding the task or joking around, the teacher can simply re-run the task. Figure 3.6 shows the output of our system for the birthday season test of Experiment 2. And Figure 3.7 shows the output of our system for the birthday month test of Experiment 2.

Experiment 3

In this experiment we asked students to highlight particular geographical locations on a projected map using the laser pointer. The teacher manually labelled the boxes

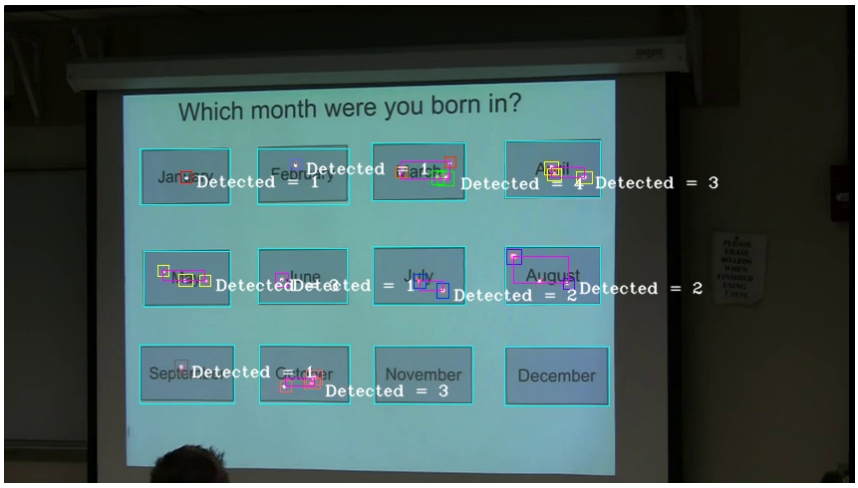


Figure 3.7: Output of our system tallying answers to a multiple choice question.

of where the votes were expected to be (i.e., the continents of the world). We then asked the students which continent they would like to visit. Our system was able to accurately cluster all of the students dots within a particular region, and further, was able to find the centroid of the average region within each continent. Figure 3.8 shows the resulting output of our system running a geographic task.

3.2.3 Interview

We surveyed students to find out how they felt about the experiments, and were particularly interested in what they thought about the use of laser pointers as a pedagogical tool. As is consistent with what was found by Zdravkovska et al. (2010), students' reactions were mostly positive, with a majority indicating that they found the system a fun way to engage with the materials presented on the slides or the whiteboard. Specifically, students were asked whether the laser pointer was an effective way of giving feedback to the lecturer. Most of the students agreed that it was, although a few raised concerns about the increasing difficulty of finding their dots in a larger class. The teacher found it easy to start and use the system and to integrate it into the presentation of the material. For the simplest tasks, the teacher mentioned that it was probably not necessary to have an automated system

Figure 3.8: Output of our system tallying answers to geographical question. Our system makes it easy for a teacher to add regions (the light blue boxes) to a slide.

at all, since it was easy to discern the voting simply by looking at and counting the clusters of dots. However, for tasks with more choices, the teacher indicated that it was helpful to have the system automatically cluster and count the dots in order to interpret the results. The teacher also noted that although the user interface was straightforward to use, the labeling of the clusters was cluttered and hard to read when there were more than a few clusters and when the labels were overlaid on top of images.

3.2.4 Conclusion of 3.2

Like other ARSs, our system was not designed to replace tests and quizzes, which are carefully graded to measure the class's understanding, but rather as an informal way to poll the classroom as well as to encourage students to collaborate with one another. We were satisfied with the accuracy of our results with these simple tests, and were further encouraged by the fact that students found laser pointers to be engaging and easy to use. We believe that our system offers a low-cost, easy-to-calibrate mechanism for gathering feedback from students that may rival more complicated and/or expensive ARSs. We are especially interested in exploring

other gestures (besides pointing) that provide types of feedback that would be more difficult (or impossible) to gather in other systems. These include circling an entire area (rather than pointing on a particular spot), indicating a range through wiggling the laser pointer, and indicating a path through a slower, more methodical tracing motion. Another aspect of our system that could be improved includes implementing a labeling algorithm to make it easy to read the results of clustering the dots. Finally, we plan to conduct a larger study using real-world tasks from teaching materials for an existing class. In particular, we plan to study the effectiveness of laser pointers versus clicker technologies in terms of student engagement and retention of knowledge.

CHAPTER 4

Pointing Gestures

Laser gestures inform us about the focus of the lecturer on the screen. They are particularly useful and helpful when the projector screen is high above the physical reach of the lecturer. When this is not the case, lecturers may use their arms or pointing sticks to point at the projector screen. However, understanding gestures, as done with the Microsoft Kinect, requires some understanding of the 3D scene. This is not available for the majority of lecture videos available online. It is not, however, necessary to understand the scene in order to find the gestures. In particular, it may not even be necessary to have a sophisticated model of a pointing object to identify pointing gestures. In this section, we present an approach for identifying pointing gestures by taking advantage of shadows created by the lecturer. Moreover, this analysis requires almost no model of shape or color of the lecturer nor the slide, nor about the arm and fingers postures used during the gesture.

In the following section, we present a system that can find the pointing gestures with promising preliminary results. The authors of this manuscript are Qiyam Tung, Alon Efrat, and Kobus Barnard.

4.1 Pointing Hand Gestures

Our goal is to take lecture videos and augment them so as to emphasize gestures and improve comprehension. With many universities offering free online courses and the advent of massive open online courses ($MOOC$), many more lecture videos are becoming available. A significant amount of these videos have a recording of the lecturer physically interacting with a whiteboard or projector screen.

During lecture, a presenter may use pointing gestures to indicate regions of importance within the whiteboard or projector screen. Knowing where the lecturer

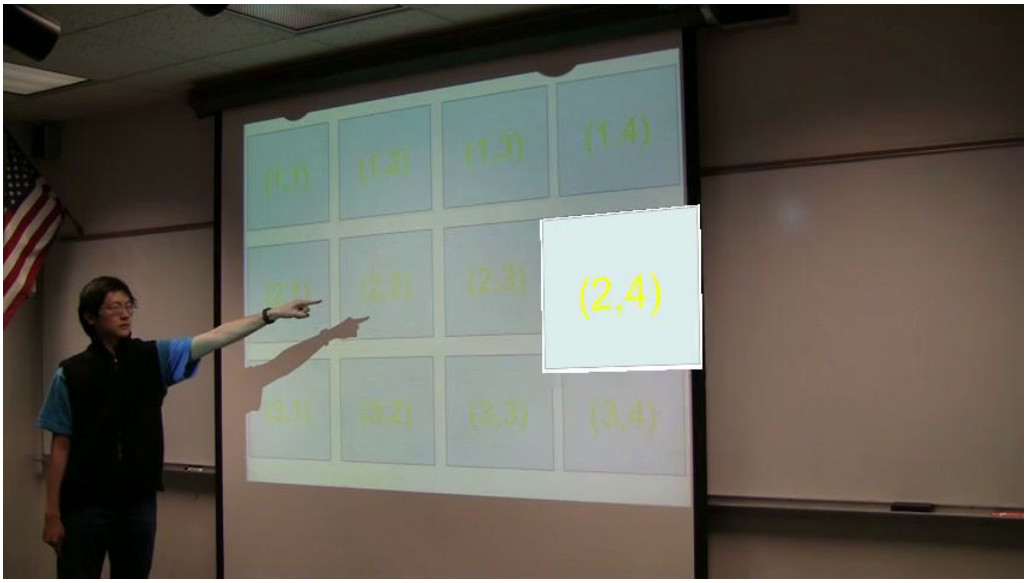


Figure 4.1: An application of finding the pointing gesture. Once a text bullet is identified, it can be backprojected and magnified into the video to improve legibility.

is pointing to is important for extracting useful information, such as the text or image that is being referred to. For example, the text from the region can be used as potential video tags or can be magnified to improve the viewing experience. Knowing when the lecturer points to a bullet is also useful in tasks such as searching and indexing. In the SLIC project, videos can be searched by their presentation slides (Fan et al. (2007)). By identifying when and where a region of interest is being pointed at, the system can further refine search to when a text or image is being referenced by the lecturer, as seen in Figure 4.1.

For simplicity, we define the exact point the lecturer is pointing to as the *reference point* on the screen. Our work of finding the reference point can be applied to videos that use both whiteboards and electronic slides. However, electronic slides, for which there is significant work on identifying content (Wang et al. (2007)), will further allow us to identify exactly what information is being pointed at. We therefore focus this paper on videos with presentation slides and solve the problems of (1) identifying the reference point and (2) extracting semantic information at that region.

Simply locating the pointing stick or hand in a 2D image is not enough to find

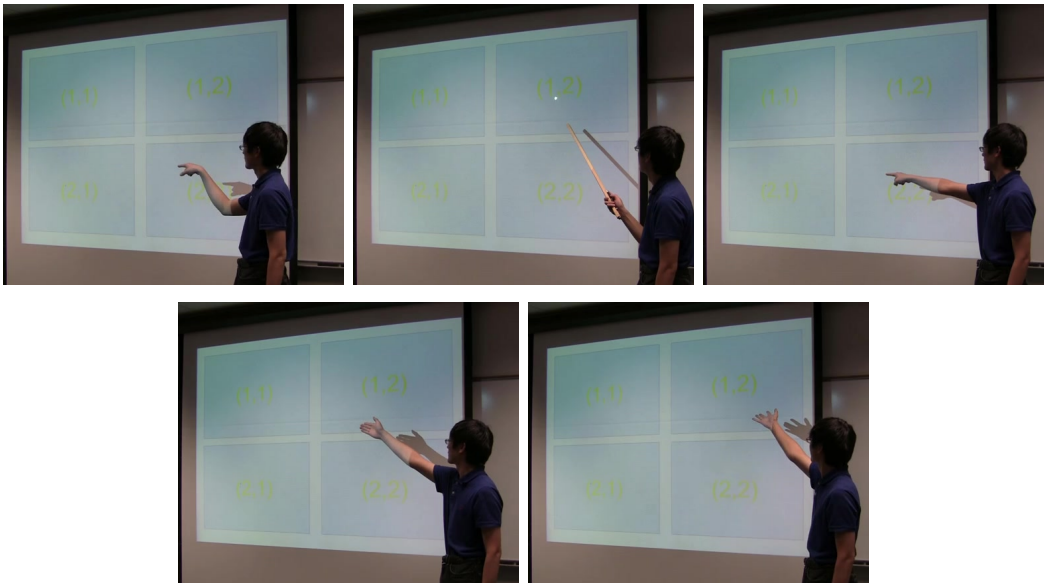


Figure 4.2: An example of all 5 gestures. From left to right: pointing with an arm with a bent elbow, a stick, a single finger, a closed palm, and an open palm.

the reference point. That is, one cannot assume that the point is in physical contact with the lecturer's pointer. It may be located elsewhere on the slide, far from the pointer. One approach to solve this problem is to reconstruct a 3D model of the lecturer for each frame. More specifically, knowing the direction and position of the hand and finger provides a direct way to find the point on the projector screen. This can be done using multiple cameras and solving the correspondence problem.

However, this would require more setup and processing time and potentially expensive hardware. More importantly, this solution would be incompatible with the thousands of lecture videos already available. Our goal is not to estimate the pose of the lecturer, but rather to identify the reference point and its contents.

In fact, in this section, we show that we can identify the reference point with minimal knowledge of the pointing object and without the use of 3D reconstruction. Instead, we use the correspondence between the pointing object and its shadow to find the reference point. Our contributions are therefore:

- A simple and fast method of identifying the reference point (the point the lecturer

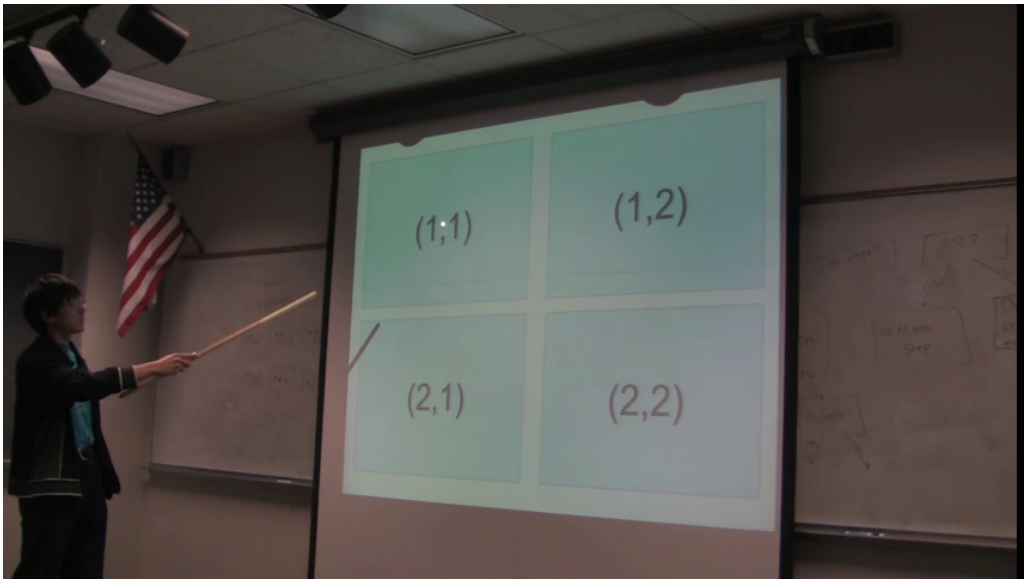


Figure 4.3: An example of a lecturer using a stick to point to a region of interest in a slide. The white dot in this region is from a laser pointer we have used to groundtruth and indicates where the lecturer is pointing to.

is pointing to).

- A robust algorithm that identifies which portion of the hand participates in the pointing. That is, an algorithm that is not limited to only identifying a finger, palm, arm, etc. (see Figure 4.2). Our algorithm does not use any pre-acquired database of images or models of the hand and is not restricted by the pose of the fingers.

4.1.1 Geometric Preliminaries

The following simple observation sets the foundation of our algorithm for pointing gestures understanding and, in particular, for determining the reference point. For simplicity's sake, we describe the theorem with the assumption that the lecturer is using a stick for pointing, but it also extends to the case of using a finger, arm, etc.

We need the following definitions first: we visualize the stick as a vector $\vec{v}_1$ emerging from the lecturer palm toward the screen. Assume that illumination is

predominantly emerging from a point source light b illuminating the screen (see Figure 4.4). Let p, the *reference point*, be the point on s the stick is pointing toward. Considering v_1 as a segment, let ℓ_1 be a line containing v_1. Let v_2 be the shadow of v_1 on the screen s, created by the light at b. Let f be the frame of the screen or viewing plane of the pin-hole camera capturing the screen, and let ℓ_1', ℓ_2' and p' be the projection on f of ℓ_1, ℓ_2 and p respectively.

Theorem 1 *The projection p' of the reference point p onto f is the intersection of ℓ_1' and ℓ_2'*

Proof: Imagine gradually changing v_1 by using a longer and longer stick, without changing its orientation. As it is directed toward s, it will eventually hit the reference point p. During this process, its shadow v_2 extends as well, and must meet ℓ_1 at p as well. Note that ℓ_2 is the intersection of the plane s containing the screen, with the plane h containing both the light source and the line ℓ_1. Hence ℓ_2 is a line, and $p = \ell_1 \cap \ell_2$.

Next let q be the focal point of the camera, and let f be its viewing plane. Let h_1 (resp. h_2) be the planes containing q and ℓ_1 (resp. ℓ_2). Their intersections with f define the line ℓ_1' (resp. ℓ_2') whose intersection point p' must lie in both h_1 and in h_2, hence in their unique intersection point in f, which must be p'. **QED**

In other words, once the stick and the shadow have been identified, it is sufficient to simply find the intersecting point of the lines ℓ_1' and ℓ_2' created by the stick and shadow in the video frame, respectively. If the lecturer only uses hands, arms, or different fingers for pointing, we have to identify equivalent segments in the pointing stick and shadow. Figure 4.2 shows the the variety of gestures we used in our experiments. Unlike the stick, the pointing direction is not always consistent along the arm. Robust and general methods for these problems are the major contributions of this paper.

Note that because the projector's light is quite luminous, the effects of other light sources are negligible. The theorem allows us to work in the image domain only. It is sufficient to identify the stick and its shadow, as two segments in the

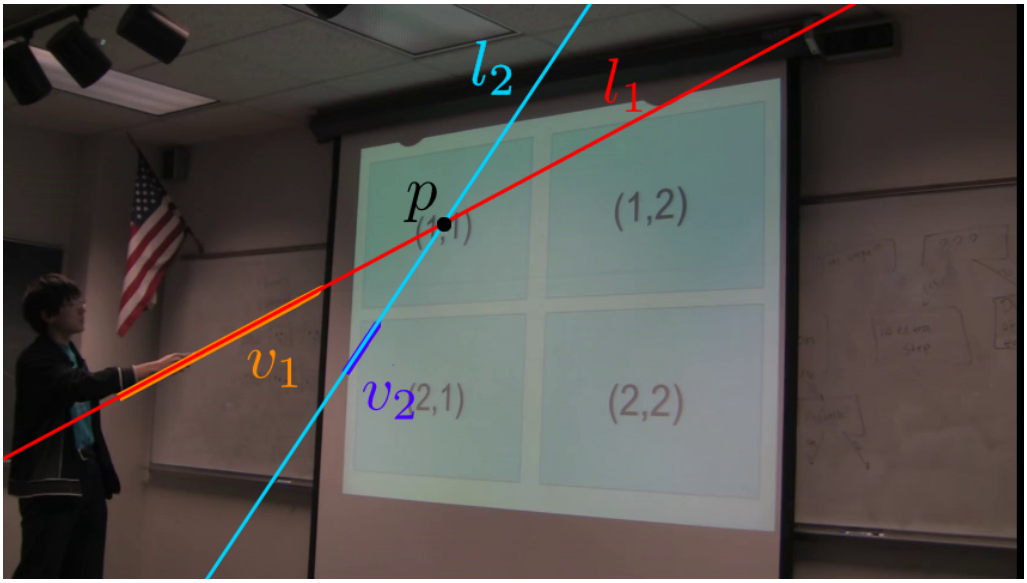


Figure 4.4: A lecturer points to the *reference point* p on the screen. This point is also the intersection of the line ℓ_1 containing the pointing stick (or the arm) and the line ℓ_2 containing the shadow of ℓ_1. This fact is used by our algorithm to find the point p' which is the projection of p on the camera image plane.

image, extend them to lines, find their intersection point and backproject this point to screen coordinates.

4.1.2 The algorithm

From Theorem 1, we can construct the following outline for finding the reference point. We present the high level description of the algorithm first and then show how each part is obtained. Here $f(t)$ represents the frame obtained by the camera at time t. $s(t)$ represents the slide shown in the video at time t.

1. Generate the median images for each slide s at time t, $\bar{f}_{s(t)}$. Then perform background subtraction and thresholding on $f(t)$.
2. Separate foreground pixels into shadow and non-shadow images.
3. For both shadow and non-shadow images, find potential candidates for fingertips or the end of a stick, which we denote as *extremal points*. A more detailed

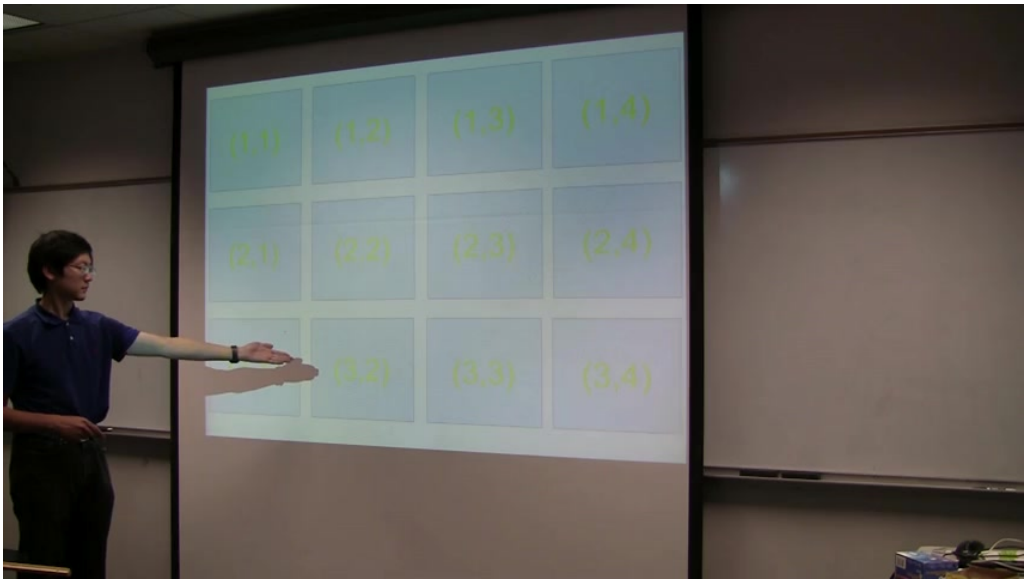


Figure 4.5: A lecturer points to the screen with his palm.

definition will follow.

4. Create lines and find the intersection points of all pairs of lines from the shadow and non-shadow images. If the intersection point of the two lines is at a reasonable location (i.e. within the slide), then it is considered a candidate for identifying an event.

Background subtraction

Because the projector screen often changes slides within a lecture, we create a median image for each sequence of frames where the slide is static. Using Fan et al. (2007), we can find these sequences. To create a median image, we sample 21 frames of the sequence and take the median value of each red, green, and blue component of the pixel independently.

Subtracting the background will reveal the pointing object and its shadow as the lecturer tends to move it about. It will also consider the presenter as part of the foreground if he or she is in the video frame. We create a bounding box around the slide slightly longer (50% wider in our experiments) to capture parts of the pointing

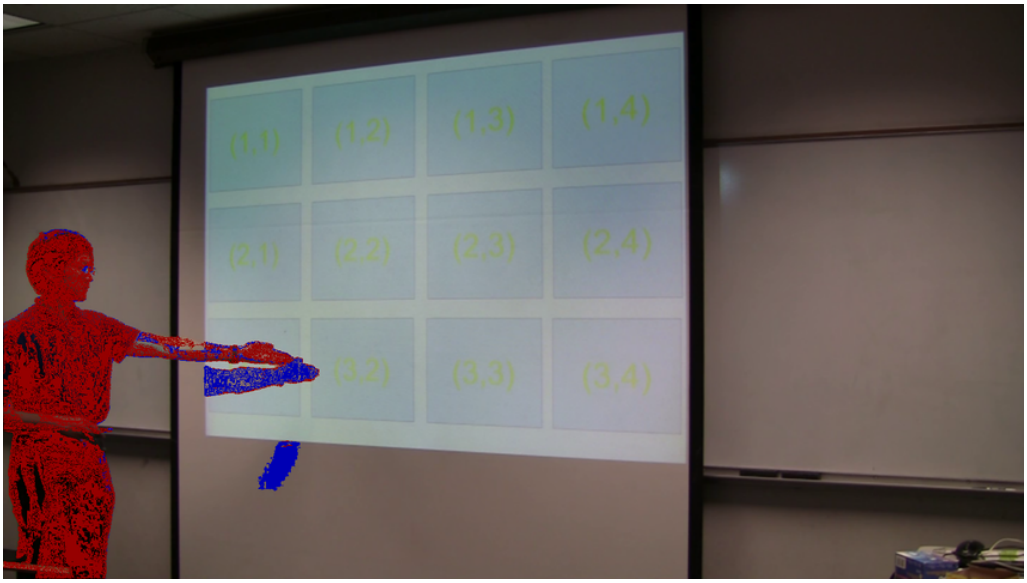


Figure 4.6: The same frame as Figure 4.5, after background subtraction and finding shadow and non-shadow pixels. The foreground pixels that are shadows are colored in blue while the rest of the foreground is colored red.

object that may lie outside the slide region.

Separating shadow from non-shadow

In order to get meaningful line segments, we need to have some notion of what a shadow looks like. The algorithm filters out all pixels not within a chromaticity range, which was obtained from manually sampling the colors of a shadow from a frame. This filtering process divides the foreground into two different images: one with shadows and the other without.

Finding the extremal points

Once the two images are created, we search for line segments that are likely to be representative of the pointing object or its shadow. We apply to each image separately.

As the foreground consists of the pointing object and its shadow, one approach is

to simply use a standard line-fitting algorithm to find the lines and their intersection. However, since we are finding pointing gestures that vary greatly in shape (as seen in Figure 4.2), we need a method that is better suited for the task.

We use the following intuition to fit lines that are *robust* to the shape of the point object. Specifically, we need to find "long and pointy" regions. Aside from the stated assumption, we do not assume any specific kind of shape. Objects that fit this description include:

- A pointing stick
- Lecturer finger(s) with a portion of the palm
- Lecturer arm (till the elbow)
- Lecturer arm (till the shoulder)

We first need to find candidates for the tips of the pointing object. Intuitively, this can be found by considering looking at pixels that are the leftmost or rightmost, depending on where the lecturer is pointing. To determine whether the lecturer is pointing to the right or left, we use our knowledge of the slide's location by computing the homography using techniques from Fan et al.. If the foreground's center of mass is closer to the right side of the slide than the left side, then the lecturer is likely to be pointing to the left. The homography allows us to compute how close the center of mass is to the slide, and therefore allows us to determine whether the lecturer is pointing to the right or left.

Without loss of generality, we assume that the lecturer is pointing to the left for the rest of this paper.

Longest Line Segments

Having determined the direction, we then find all points that are locally leftmost. That is, a point is the locally leftmost point if among its eight neighboring pixels there are no foreground pixels to the left of it. We then group all vertically contiguous leftmost points and pick the median from each of those groups. These points now represent potential candidates for the tip of a finger, hand, or stick.

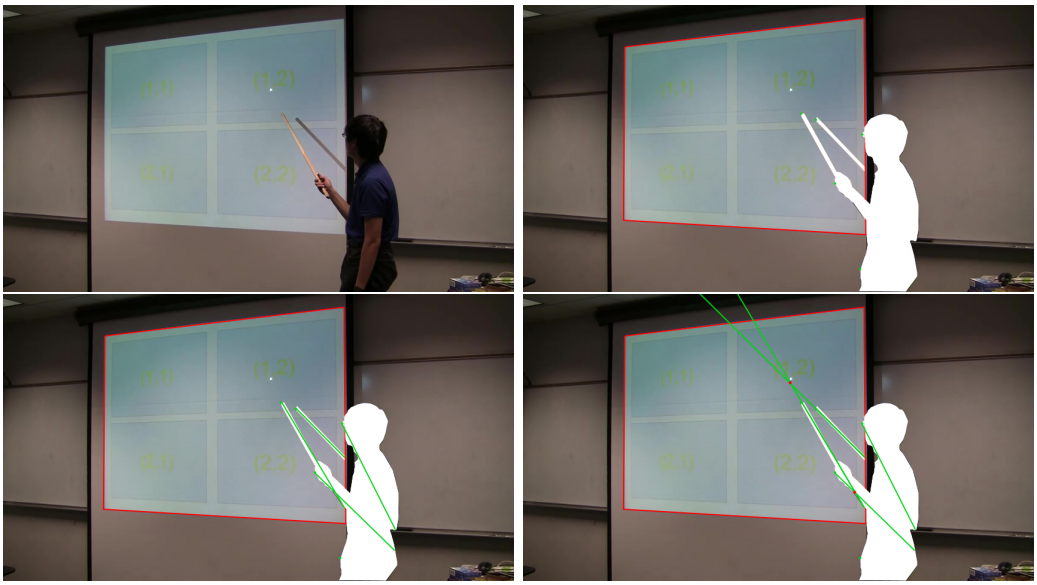


Figure 4.7: The reference point is found by searching for long line segments and their intersections. The algorithm to find these segments works by finding the locally leftmost points in the foreground (top right) and, for each leftmost point, finding the longest line segment from that point (bottom left). Each resulting segment then extended to a line and is a candidate for a pointing object or its shadow. The intersection points of all pairs of lines each contribute a vote to each bounding box and the box with the most votes is considered the reference point. This figure is an illustration of the main ideas of the algorithm and is not actual output from the program.

For each of these points, we find the longest line segment. To find it, we select the segment with the longest length out of all the segments formed by the leftmost point and *boundary pixels*, which are pixels that have at least one non-foreground pixel among its eight neighbors.

It is possible that the line segment formed by the leftmost point and the boundary pixel may not lie within the foreground, such as forming a line from the finger to the knee. We therefore discard the segment if there is a contiguous non-foreground segment of the line longer than 20% of the segment's length. The process of the algorithm is illusrated in Figure 4.7.

Selecting Line Candidates

The aforementioned process will create many lines, many of which may not lie along the arm. Although the pointing object such as an arm tends to have long line segments, a long line segment need not come from the arm. For example, the segment can be formed from the shoulder to the leg of the lecturer. To obtain reasonable line segments, we score each line based on its length as well as how far it is to the globally rightmost point. The motivation for this is the same as choosing the leftmost points when finding line segments: the pointing objects and its shadow tend to be to situated to the left of the lecturer.

The score s for each line is then $s = 0.5l + 0.5e$, where l is the length of the line normalized to the largest line in the foreground and e is the normalized distance of the segment's leftmost point to the global rightmost point.

For our experiments, we chose the 8 lines with the highest scores for each of the foreground images. To determine the reference point, we find the intersection points of all possible pairs of lines between the lines derived from the shadow and non-shadow images.

But in order to find a meaningful result, we must first identify the region of interest within the slide itself.

4.1.3 Identifying the Region of Interest

To give practical application for finding the point on the scene, we use the following technique to find which bounding box is being identified.

We use the techniques outlined by Tung et al. (2011) to partition the presentation slide into meaningful portions. Specifically, the presentation slide is divided into rectangles that bound a text or image. We call these *bounding boxes*. Examples can be seen in Figure 4.1, where it has been magnified. It should be noted that a meaningful distance between boxes can only be computed if the homography is known. Otherwise, the actual distance from one bullet to another may be skewed by perspective. We make use of the homography obtained from techniques such as

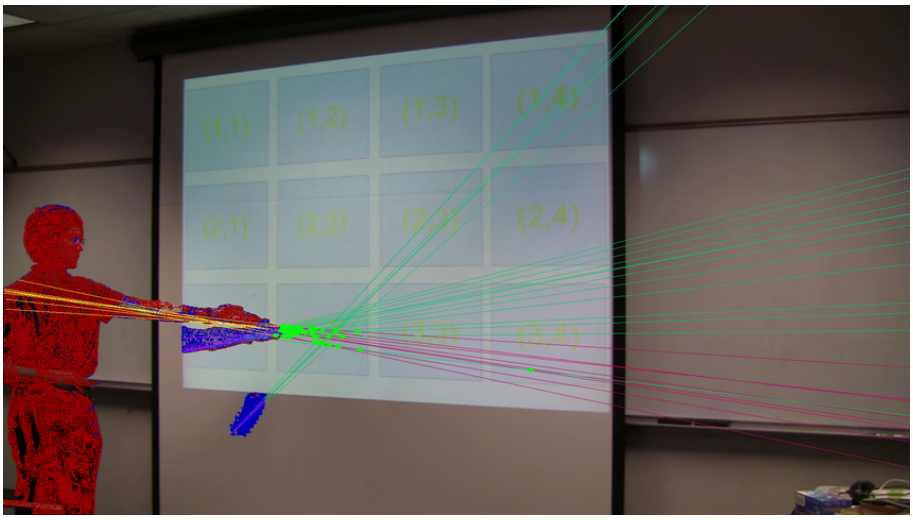


Figure 4.8: After the shadow and the non-shadow segments have been identified, we fit lines to the corresponding segments. Non-shadow lines are colored magenta and shadow lines are colored blue-green. The intersection points, colored green, are used to vote where the lecturer is pointing to within the slide. It is worth noting that in this image, both the shadow from the projector light and the shadow from the room lights (bottom blue segment) were correctly identified, which improves the identification of the reference point.

those used by Fan et al..

Given an *event*, time intervals in which the user is pointing to an object in the slide, the algorithm computes the intersection of every pair of lines for the frame in the interval.

The intersection points are converted to slide coordinates using the homography. Then the bounding box that it lies closest to gets a vote. If the point is too far outside of the slide, then it is also removed from the list of intersection points. We set the distance to be a quarter of the length of the slide. The process is repeated for all frames of an event and the box that has the most votes for the event will be considered the region of interest.

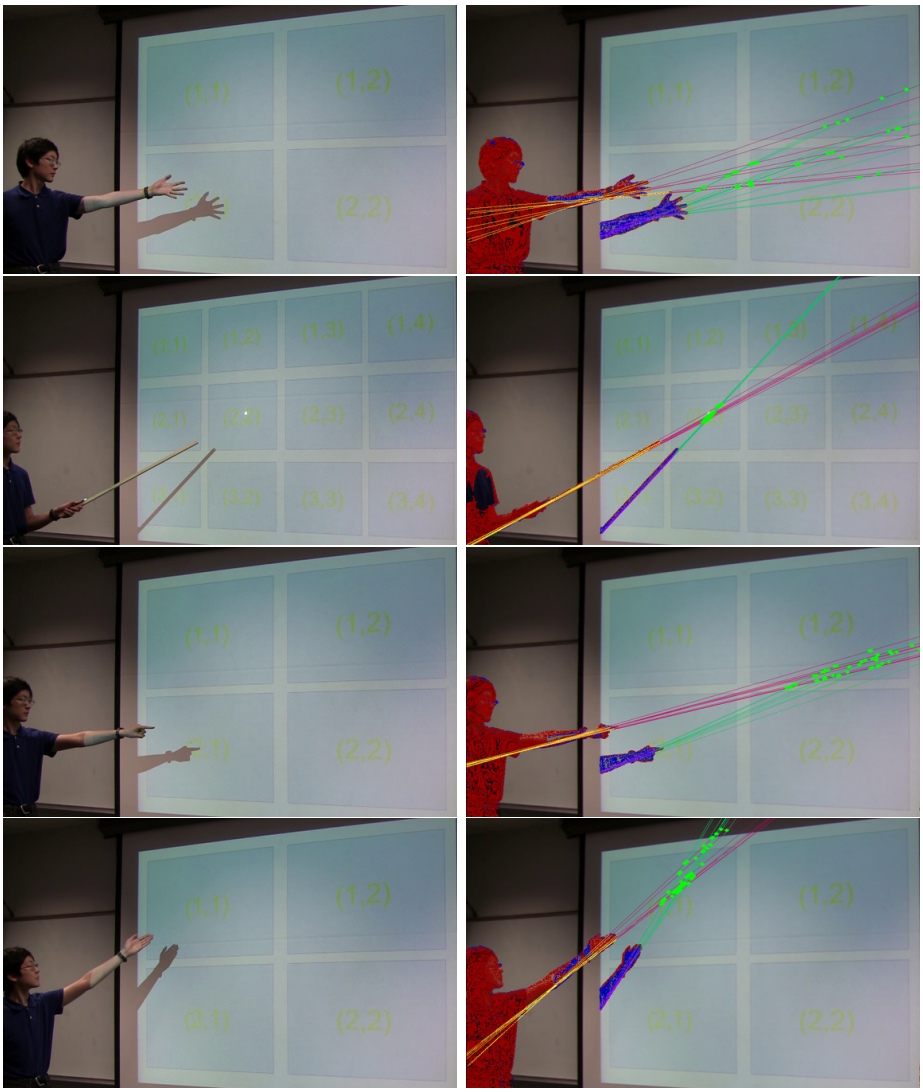


Figure 4.9: This figure shows 5 frames processed using our algorithm. These were processed on high resolution images to illustrate how our algorithm works. In the first two frames from the left, the lines are fit to the locally rightmost points, such as the lecturer's fingers. The same algorithm can be used to find the tip of a stick, pointing finger, or the end of a closed palm (the 3 last bottom images). Note that applying linear regression would result in steeper lines because the red foreground's mass is mostly distributed in the lower half of the image.

Accuracy		
Gesture	Correct	Number of events
Stick	19	20
Single Finger	5	8
Closed Palm	14	20
Open Palm	6	8
Bent Elbow	5	14
Total	**49**	**70**

Table 4.1: Gestures in the video

4.1.4 Results

In our experiment, we created a video where the lecturer used multiple gestures to point to the reference point. The bounding boxes partitioned the slide into rectangular regions. There were two slides used, which can be seen seen in Figure 4.2 (4 large boxes) and Figure 4.1 (12 small boxes). The lecturer pointed to a number of boxes in each slide. Our algorithm identified the correct bounding boxes in 49 out of 70 events, giving an accuracy of 70%. Sample results can be seen in Figure 4.9.

Out of the 40 events for the slide with 4 boxes, 29 were correctly identified. For the second slide, which divided the slide into 12 boxes, the algorithm correctly identified 20 out of 30 events.

A more detailed look of the results can be seen in Table 4.1. A gesture made with a stick was almost always correctly identified due to the fact that it is long and thin, which makes longest segments found to be relatively consistent.

For the hand gestures where the arm is relatively straight (Single Finger, Closed Palm, Open Palm in the table), the average accuracy was 69.4%. Despite the variations in the shape of the hand, the algorithm still managed to perform decently. However, the bent elbow got an accuracy of 35.7%. The main cause for this is because the direction depend heavily on the orientation of only the hand. As such, long sgements were not a particularly good indicator of the direction of the pointing

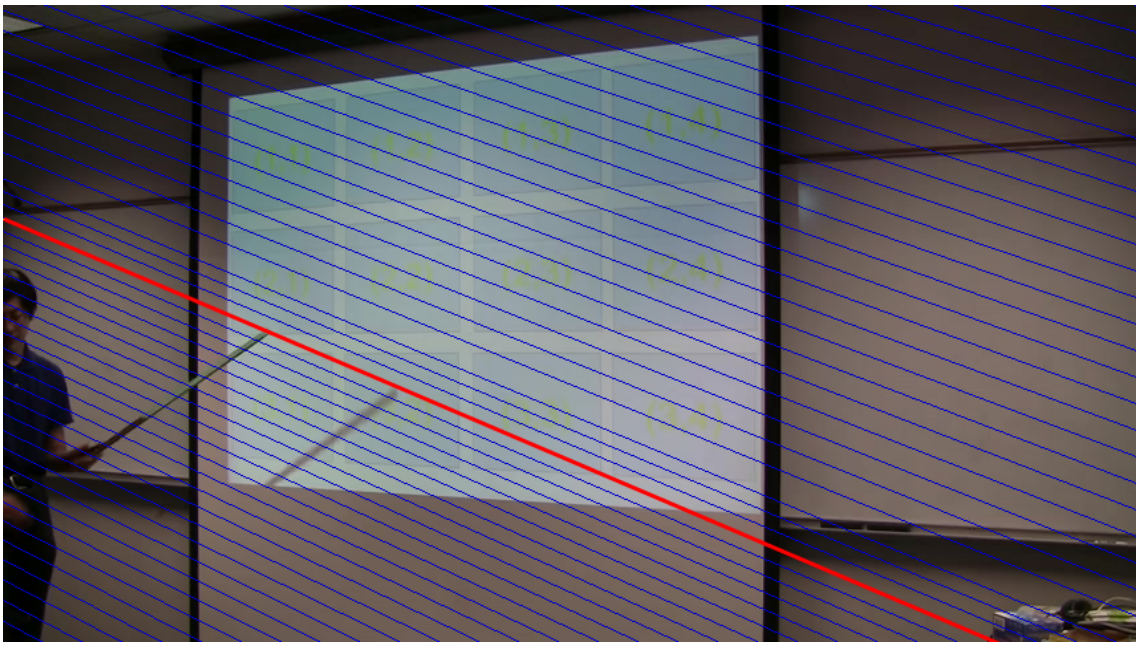


Figure 4.10: The lines in the image represent the epipolar lines, which all intersect at the epipole. Put another way, the rays of light are emitted from the projetor (outside the frame). In the context of identifying pointing gestures, it helps in limiting the search of matching long line segments to lie along the epipolar line. The red line shows the epipolar line for the tip of the stick. Note that the tip of the stick lies on the same line as the shadow of the tip of the stick.

object (see Figure 4.2).

The algorithm was run on a machine with a 2.3GHz AMD Athlon processor. On average, it processes a 854×480 frame every 3.7 seconds.

4.1.5 Using the Fundamental Matrix

In theory, the results can be improved through the additional use of the fundamental matrix. The fundamental matrix is the algebraic representation of the relationship between the projector and the video camera. In the context of lecture videos, it relates an object with its shadow. That is, the fundamental matrix says that the shadow of an object must lie along the ray from a projector (see Figure 4.10). For

example, the shadow of the end of a stick cannot lie just anywhere in the image but must lie along the epipolar line. For a review on epipolar geometry, see Section A.1. It is known from Hartley and Zisserman (2003) that the fundamental matrix can be computed if the epipole (the location of the projector center in our case) and the homography are known. To find the epipole, one needs to find a point matches between an object and its shadow projection.

In order to make use of this, we would need to use frames where the identity of the stick and its objects are fairly certain (i.e. where the score is high) and then use the fundamental matrix to find better matches in frames where it is not as clear.

We have done some preliminary experiments using the fundamental matrix and while it did remove incorrect matches, we did not see a significant improvement in accuracy in terms of finding the correct bounding box.

4.1.6 Conclusion of 4.1

We have developed an algorithm that processes a video reasonably fast to find the location of the reference point. In particular, we have shown that, once the background image is processed, we can determine it entirely from a single image.

CHAPTER 5

Further Applications

In addition to the magnification, we can take advantage of the knowledge of the scene to reduce power consumption. In Section 5.1, we show that by selectively dimming unimportant regions such as the lecturer and scene background, mobile devices that use OLED displays can save a significant amount of energy.

This section was published in the 10th International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (MOBIQUITOUS) in 2013 (Tung et al.).

5.1 Energy-savings on Mobile Devices

We present a context-aware system that simultaneously increases energy-efficiency and readability for educational videos on smartphones with OLED displays. Our system analyzes the content of each frame of the video and intelligently modifies the colors and presentations of specific regions of the frame to drastically reduce display energy consumption while retaining relevant content of the lecture video. We achieve this by leveraging the mapping between frames and electronic versions of slides used in the lecture. This enables separate manipulation of the slide area and the background. Further, since the slides can themselves be analyzed for content (e.g. text and images within) this approach provides substantive control over energy use and user experience. We evaluate the system using extensive energy measurements performed on phones using two different display technologies. Our method was able to reduce energy usage up to 59.2% of the energy used by the display which amounts to 27% of the total energy used by the device.

We are becoming more dependent on smartphones and expect long battery life, despite steady increases in features and performance. As more smartphone users are

spending their time on multimedia applications (Smith (2010)), such as watching movies, the demand for brighter displays that can be visible even in bright light is growing. Among the videos that are watched on mobile devices are educational videos available on-line (mit (2014); ber (2014); yal (2014); cou (2014); uda (2014); edx (2014)). Large and high resolution displays of modern smartphones have made it easier to access and watch educational videos lectures in situations where using a regular desktop or laptop is inconvenient, such as commuting. However, being able to watch long videos depends on a reasonable battery life.

Displays in smartphones often account for a significant amount of the total energy consumption, making them one of the primary targets for energy optimizations. In particular, organic light-emitting diodes (*OLED*) displays, whose energy consumption is directly related to the color and brightness of the pixels (Dong et al.), present opportunities for energy savings, such as modifying the colors and intensity of pixels in regions that are less important to the viewer.

This paper develops a system for modifying educational videos with consideration of the *content* appearing in each frame. Our techniques enable us to make selective adjustments that minimize the impact on the viewing experience while at the same time significantly reducing energy consumption. In addition, our technique has the ability to improve the readability of the slides in the video. We present two main approaches for reducing energy consumption for lecture videos. First, we utilize *backprojection* to replace portions of each frame occupied by the slide by the original slide used in the presentation. The slide color scheme is also revised to reduce energy consumption and increases readability. Second, we alter background regions of the video frame to improve energy efficiency, while only impacting the visibility of the region that is less important to the context of the video.

Hints to automatically understanding the content are obtained from analyzing slides used in the lecture and cross-correlating them with the video itself. Our techniques work for recorded videos and hence are applicable to any of the discussions, lectures, colloquiums, etc., recorded already, as long as one can access the electronic slides being used in the lecture.

5.1.1 OLED Displays

While dimming the backlight was the main method of decreasing energy consumption for liquid crystal displays (LCD), OLED displays offer greater potential for energy savings as the diodes are emissive and do not depend on a global backlight. The energy in an OLED display is directly correlated to how many pixels are on and what intensity level and color they are displaying. However, different display technologies may alter the energy consumption behavior of the display. In particular, phone displays may have different subpixel arrangement of the three primary colors, as is the case with the devices we used (Figure 5.1). The Samsung Galaxy S Vibrant uses the PenTile matrix array (PMA) technology sam (2011) and the red, blue, and green OLEDs have different dimensions. Furthermore, the pixels are laid out so that each one contains only two OLEDs. In order to generate the full range of colors, PMA displays utilize subpixel sharing. For example, in order for a pixel that does not contain a blue OLED to display blue, its adjacent pixel's blue OLED would need to be activated.

Samsung Galaxy S2, on the other hand, uses a standard RGB stripe pixel array. The layout is simpler and the three OLEDs are arranged in vertical stripes of red,

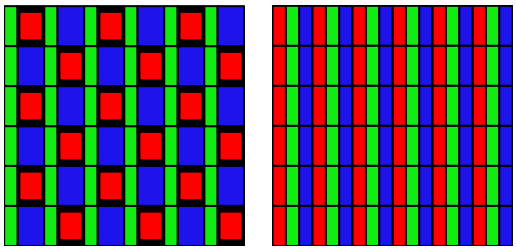
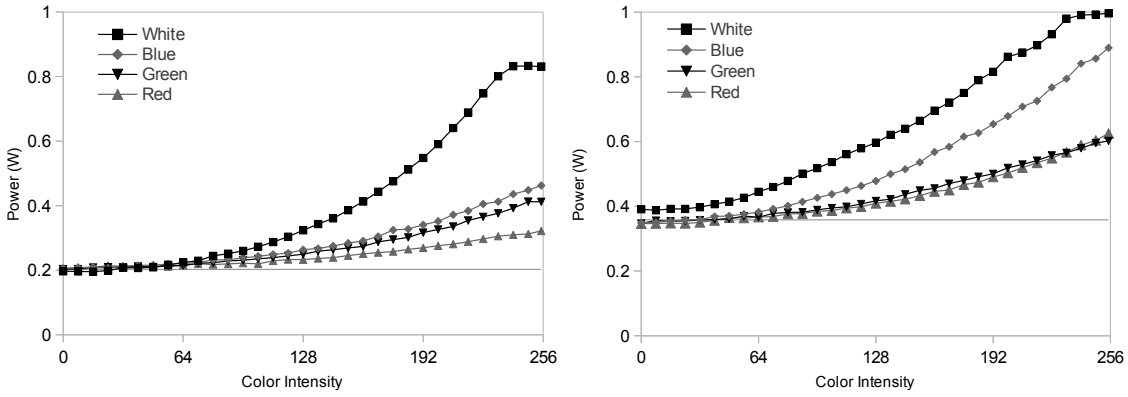


Figure 5.1: Layout of pixels of different channels for the (left) Pentile matrix and (right) RGB stripe matrix.

Phone	Galaxy S	Galaxy S2
OS	Android 2.2	Android 2.3.5
Color depth	24 bit	24 bit
Display size	4 inches	4.5 inches
Display type	PenTile	RGB Stripe
Resolution	480×800	480×800
CPU	1GHZ	2×1.5 GHZ
Memory	512 MB	1 GB

Table 5.1: Smartphone Specifications



(a) Galaxy S (b) Galaxy S2

Figure 5.2: Measurements of the power draw of the three primary colors for all sRGB values

green, and blue and hence do not require subpixel sharing. The specifications of the two phones used in our experiments are shown in Table 5.1. Despite the differences, the blue channel consistently consumes the most energy in both devices, as seen in Figure 5.2. Therefore, in terms of color manipulation, our methods either dim all colors equally or dim out the blue channel. The choice depends on the viewer's preference and content of the lecture.

5.1.2 Improving Energy Efficiency

Energy optimizations we are proposing are tightly related to the layout of a video frame. A typical frame consists of several regions (see Figure 5.3):

1. The *non-slide background* consists of the audience, walls around the screen, and any other regions outside of the projector screen.

2. The *slide region* contains elements that carry vital information, such as text, figures, pie charts, graphs, etc.

3. The region of the slide not containing the text or images is the *slide background*.

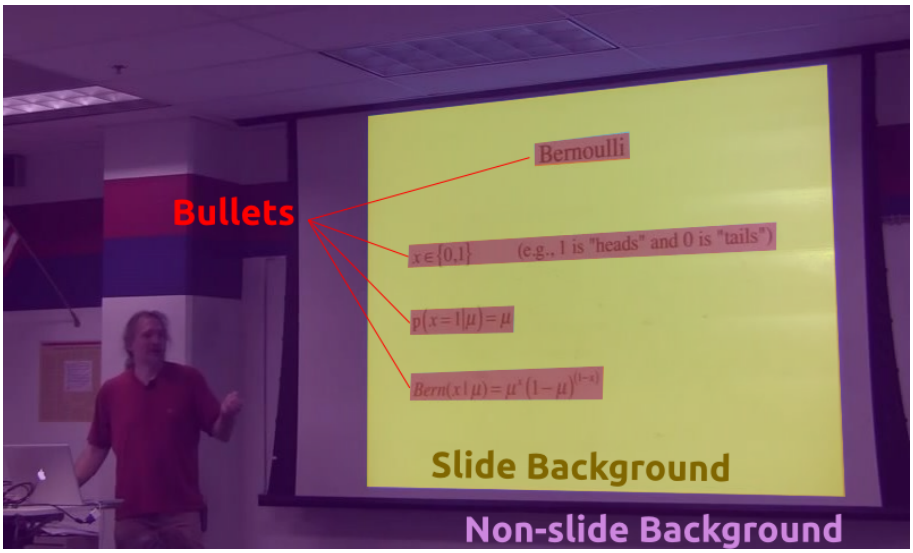


Figure 5.3: The three kinds of regions in a video frame.

The challenge is in identifying these regions in a deck of electronic slides, as well as in the video containing these slides. The energy optimizations can vary from recoloring of the whole frame to recoloring the individual regions as identified above.

5.1.3 Altering the Non-Slide Background

The slide background may contain speaker's gestures, audience reactions or interactions, etc. While a speaker's gestures may be important, the absence of a color or background all together is unlikely to prevent a user from understanding the lecture content. However, the desirability of the background depends on the person, and is orthogonal to the rest of the discussion.

To detect the background and slide regions, we first identify the slide used in each frame, along with the geometric mapping between slide and frames. We use the automatic slide detection already developed in the SLIC project (sli (2014)), which gives us both the geometric transformation, the *homography*, as well the *temporal* information, which tells us which slide in the slide deck is shown in each video frame. Once the video frame layout is identified, we can dim or alter color by removing the most power hungry color, which is blue, from the background frame, as seen in

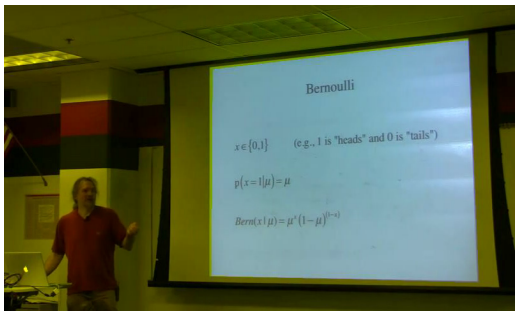


Figure 5.4: The blue channel is dimmed from the background.

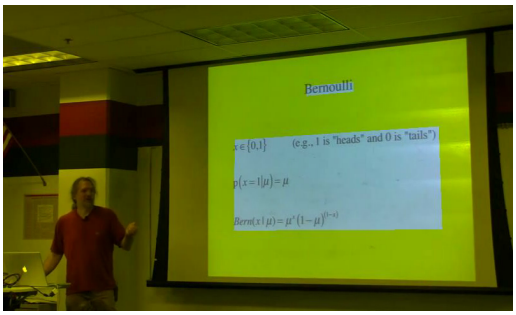


Figure 5.5: The blue channel is dimmed from the non-bullets regions.

Figure 5.4. Surprisingly, this approach is rather ineffective for saving energy, as the non-slide background area is often small and much dimmer than the slide portion of the frame. Subsequently, the slide portion of the frame is the most power hungry portion and critical for successful energy optimizations.

5.1.4 Altering the Slide Background

To further take advantage of dimming nonessential regions we need to consider the slide area that is not covered by text or images as shown in Figure 5.5. Slide text and images tend to occupy a relatively small portion of the slide and the video frame under consideration. Subsequently, by dimming or removing color from such regions not occupied by text or images, we can get significant savings.

To identify text, embedded equations, and figure regions in the slide, we rely on techniques developed in Section 2.2. We also assume we will have the homography. From this information, we can then calculate the appropriate regions in the frame. The homogeneous coordinates of a pixel in the slide, P_s, is related to its corresponding pixel coordinate in the frame P_f by the following equality: $HP_f = P_s$. H was derived by matching slide pixels in the frame to the slide, so to find the coordinates P_f given P_s, we take the inverse, giving us $P_f = H^{-1}P_s$. With this, dimming the color channel of the pixel is finally straightforward. We scan every pixel in the slide and reduce the color value of the channel if its corresponding frame pixel is outside the bounding box.

Figure 5.6: A comparison of the original (right) with the backprojected slide (left).

5.1.5 Altering and Replacing Slides

To achieve slide background modification we went through color alteration of text and images to detect text and image regions as well as slide mapping to the video frame. We can combine those techniques and replace the entire slide in the video frame with the slide optimized for energy efficiency. To generate energy efficient slides we can either do them by hand or utilize previously described techniques to automatically change text and slide background while keeping images unaltered.

To maximize energy efficiency and readability, each slide has its background modified to black and the text to white. Figures, pie charts, etc., appear in their original colors and intensities since automatic processing is not able to recolor images and guarantee that they contain original information content. Once modified, the slide is backprojected into each frame it was used by Winslow et al. (2009), after making the geometric manipulations necessary to account for size and geometry of the slide in the video frame. The comparison of before and after optimization is in Figure 5.6.

While backprojection can significantly improve energy efficiency as well as readability by improving image quality, it can also hide events that happen on the

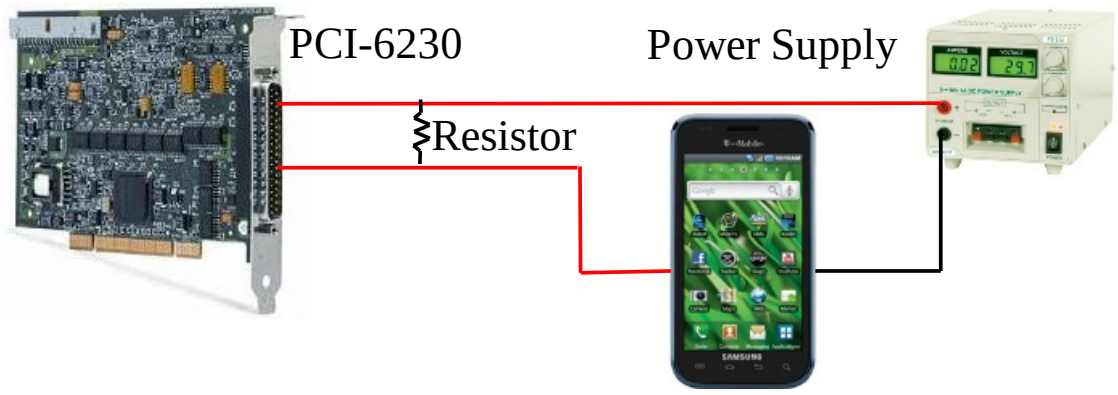


Figure 5.7: Measurement environment

projector screen, such as the lecturer pointing (with hands or laser pointer), slide animations, and videos embedded in the slides. This is due to the backprojected slide overlaying the original slide in its proper location in the video frame. For example, if the speaker walks into the region of the slide, he or she would be hidden by the slide due to the fact that it is drawn on top of the video frame. This may distract the user and is a reason why one may choose to simply dim colors instead of modfiying and backprojecting the slide instead.

5.1.6 Methodology

All videos have a resolution of 800x480, that corresponds to the resolution of the smartphone displays, and were compressed with `ffmpeg` using the h264 codec with a frame rate of 29.97 frames per second.

We evaluated the energy consumption of the proposed mechanisms using a National Instruments PCI-6230 Data Acquisition card (NI PCI-6230) and measuring software. The phone was connected directly to the power supply, set to 4.15V.

To emulate the battery, we reused electronic components from the battery pack and wired them directly to the power supply. The current draw was calculated by capturing the voltage drop across a 10mΩ current sense resistor wired into the phone's connection to the power supply (see Figure 5.7).

Video	Length [min]	Average Intensity of		
		Red	Green	Blue
Barnard	4:00	86	101	105
Berman	3:15	111	139	132
Chaves	4:00	24	119	105
Coates	4:00	89	93	145
Tung	5:27	50	80	94
Rozenblit	3:15	120	114	114

Table 5.2: Video Statistics

5.1.7 Results

The phone was connected directly to the power supply, instead of battery, for current measurements. To make the measurements reproducible, we turn off all wireless signals as well as automatic brightness adjustment. We evaluated our proposed mechanisms on the two phones, Samsung Galaxy S and Samsung Galaxy S2, over six lecture videos. We selected videos with varying lighting conditions, camera angles, and the amount of screen space that the lecture slide occupies versus background, as well as color mixes and Table 5.2 shows basic statistics such as video length and the average color composition in the video.

Efficiency of Altering the Non-Slide Background

Figure 5.8 shows the frame distribution between non-slide background (BG), background of the slide (Slide Remainder), and the text or image areas of the slide (Text/Image Boxes). The slide covers roughly 50% of the total area in the video, which is not surprising as the slide has to occupy a significant portion of the screen so that the text is large enough to be legible. The figure also shows that the text and images only take up 15% of the screen real estate on average, which means that only a fraction of the energy needed to play the original video is necessary to display the text and images.

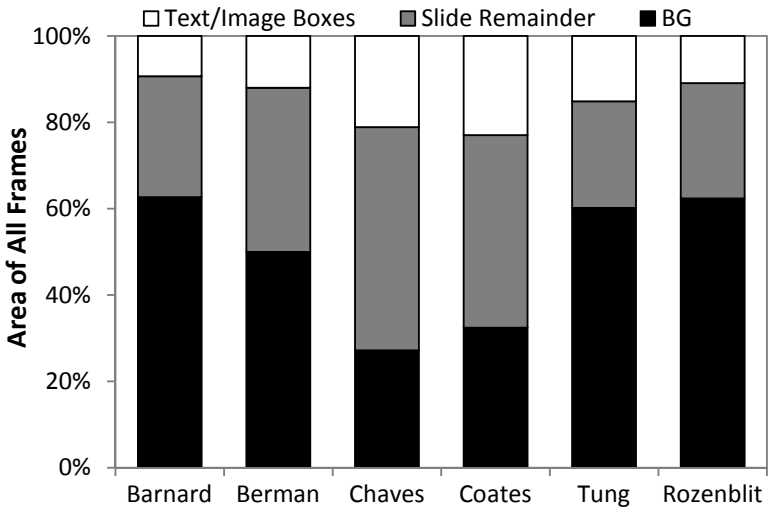
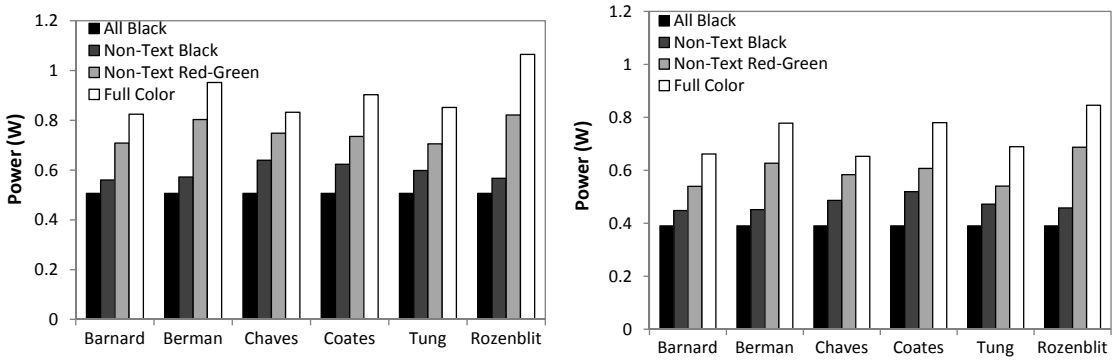


Figure 5.8: Average area distributed among a slide, slide text and images, and the background areas.

Blacking out the non-slide background corresponds to removing the energy consumption from 50% of the video frame, on average. However, the resulting energy savings are only 23.4% and 19.9% on the Galaxy S and S2, respectively. The modest savings can be accounted from the fact that the non-slide background is typically much darker that the slide area since the slides are displayed with a high power projector on a white screen and the room is dimmed to provide better slide readability.

Efficiency of Altering the Slide Background

Figure 5.10 shows progressive stages of dimming the slide background by reducing the RGB values by 20%, 40%, 60%, and 80%. The corresponding energy savings on Galaxy S2 are 20.0%, 36.2%, 44.2%, and 50.1% of the display energy usage. We observe that the energy savings come at the expense of alteration of the original image. While removing colors can make a lecture video seem unnatural, this particular transformation preserves the chromaticity of the original background but reduces the intensity, putting focus on the relevant part of the lecture, similar to other work (Iyer et al. (2003); Harter et al. (2004); Wee and Balan (2012)).



(a) Galaxy S (b) Galaxy S2

Figure 5.9: Energy consumption after dimming less relevant regions.

Figures 5.9a and 5.9b show the average power demand of the entire phone for the combination of the display dimming techniques: video with all frames completely black (*All Black*); the video where only text and image regions are displayed (*Non-Text Black*); the video where we only remove blue, the most power hungry color, from the entire frame except the text and image regions (*Non-Text Red-Green*); and the original video without any alterations (*Full Color*). The upper bound on energy savings is attained by setting the color of the entire screen to black for the duration of the video. Playing such a video with all frames being black on the Galaxy S and S2 show the average upper bound on savings of 43.4% and 46.4% of the total energy needed to play the original video, respectively. For the remaining results, we will cite the savings in terms of the display energy, which is calculated by subtracting the energy to display a black video from the original video.

Non-Text Red-Green saves 37.1% to 39.8% for the Galaxy S and S2, respectively. The savings are significant, especially when considering the diversity of the color distribution in the videos (see Table 5.2). In cases where the slide background is white, such as in *Rozenblit*, savings can increase to 19% on the Galaxy S and 22% on the Galaxy S2. The other factor in such significant savings is also due to the fact that some videos, such as *Rozenblit* and *Berman*, have long periods where no slide was shown.

Figure 5.10: Images whose slide background is dimmed by 20%, 40%, 60%, and 80% from left to right.

While the blue channel requires the most power, red and green combined still constitute a significant portion of energy. The only exception to this is the *Coates* talk due to the fact that the slides shown have a blue background (Table 5.2). Dimming out all colors (*Non-Text Black*) from the slide and non-slide background can reduce energy consumption to levels close to an entirely black video (*All Black*). In general, dimming out colors from non-bounding boxes tends to save a lot of energy as bounding boxes take up only a small percentage of the screen real estate.

Efficiency of Altering and Replacing Slides

Figure 5.11 shows the average power demand for the Galaxy S2 and the combination of the video alteration techniques: the video with all frames completely black (*All Black*); the video where the slide region was replaced by the entirely black side and the non-slide background was unaltered (*Black Slide*); the video where the slide

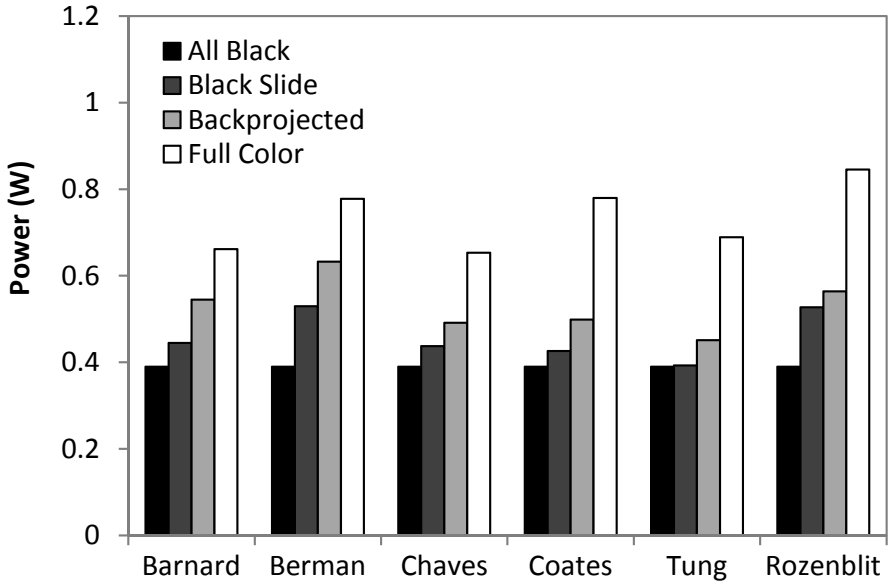


Figure 5.11: Energy consumption of videos after slide backprojection on the Galaxy S2.

region was replaced by the energy optimized side and the non-slide background was unaltered (*Backprojected*); and the original video without any alterations (*Full Color*).

All Black is the amount of power consumed by all other components except for the display. To calculate the power of only the display, we subtract all measurements by *All Black*. The *Black Slide* is the upper bounds on what can be saved by recoloring only the slide, leaving the non-slide background unaltered, and can offer 80.9% display power demand reduction. Backprojection is very efficient, offering 59.2% reduction in average display power which translates to 27.4% reduction in power demand of the entire phone. Further power reduction may be possible at the expense of lower color intensity of the text and images in the slide.

The savings for backprojecting white-on-black slides are significant because the total area of the screen taken up by bullets is less than the 15% that is taken up by the bounding boxes. Since the text was recolored rather than just keeping the original color of the whole region containing the text, the total area taken up by the text will have decreased as well. In other words, more pixels are blacked out and

that increases the savings.

5.1.8 Conclusion of 5.1

We have described a system that intelligently reduces energy consumption by replacing colors from less important regions of a lecture video. A key contribution is using slide-to-frame matching and slide semantic extraction to provide fine-grained understanding of content that can be exploited to enhance viewing and save energy. Subsequently, we have shown that removing colors from lecture videos of multiple lighting conditions is a viable method for saving a significant amount of energy consumed in mobile devices during playback. In addition, we have presented several methods to selectively remove varying degrees of different colors from portions of video frames. The resulting optimizations provided significant power reduction of displaying educational videos while minimizing the disruption of video quality by utilizing information about which areas of a video frame are the most informationally important.

5.1.9 Acknowledgments

This research was funded by the National Science Foundation under Grant No. 3138500 and the Microsoft Research Faculty grant.

CHAPTER 6

Future Work

6.1 Text-Box-Based Identification of Slides

An alternative method of identifying the slide is to use our knowledge of the arrangement of bounding boxes to determine the homography. The motivation for this alternate method is that while the SIFT keypoint/RANSAC method works, it is slow. There are many practical applications for being able to identify slides faster, such as helping with the visually impaired.

APPENDIX A

Appendix A

A.1 The Fundamental Matrix

Epipolar geometry is the geometry that arises from two camera views. It constrains where a point from one from the camera's image plane will be on the other camera's image plane. Without this constraint a point from one view could be any other point in the other view. In our problem of finding the corresponding point for tips of pointing fingers, epipolar geometry limits the search to a line. However, this geometry is more generally known in the context of 3D scene reconstruction.

Mathematically, the fundamental matrix is the algebraic representation of this geometry. One common way to decompose the fundamental matrix is in terms of the two camera projection matrices and the epipole, the image point of the first camera as viewed by the second camera. The equation is

$$F = [e]_{\times} P' P^{\dagger} \tag{A.1}$$

where $P^{\dagger}$ is the pseudo-inverse of the first view's camera matrix, P' is the second view's camera matrix, and e is a 3×1 vector representing the epipole. $[e]_{\times}$ is a 3×3 skew-symmetric matrix and is constructed using e. Given that $e = (e_1, e_2, e_3)^{\top}$, the entries of $[e]_{\times}$ is

$$[e]_{\times} = \begin{pmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{pmatrix} \tag{A.2}$$

It is constructed so that given a point, a 3×1 vector x, $[e]_{\times} x$ is the cross product between the two.

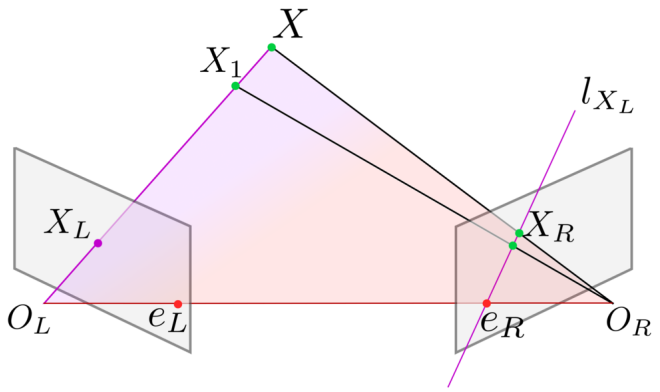


Figure A.1: The fundamental matrix describes the epipolar geometry of two views. A point in the 3D scene, X, is seen as X_L by the camera on the left, whose projection center is located at O_L. The fundamental matrix describes where X_L might be seen by the camera on the right. The corresponding point is inherently ambiguous because the point X_L could have come from X_1 or X. Therefore, the corresponding location for the camera on the right is a line rather than a point. The line l_{X_L} is constructed from the epipole e_R, the projection of the left camera on the right camera's image plane. In the absence of the fundamental matrix, X_L could be anywhere on the image as opposed to limited to a line.

The result of applying the fundamental matrix to a point, X_L, in the first view's 2D homogeneous coordinates, can be seen in Figure A.1. Briefly, it describes how the second camera views the ray of the point seen in the first camera's view. We can understand this by studying each step in the operation FX_L, which can be decomposed as $[e]_\times P' P^\dagger X_L$. $P^\dagger$ backprojects a point back to a point in 3D. Note that because 3×4 camera matrix P is degenerate, the pseudoinverse does not actually backproject the point back into the correct 3D position. But it is a point (possibly X_1) along the line between the camera center O_L and the original 3D point, X. Then it projects this point into the viewpoint of the second camera. Finally, it constructs a line by taking the cross product between the epipole and the projected point. The result is a line in the second camera's image plane of where the actual

point X might lie in the second camera's view.

In the work of finding pointing gestures, this eliminates the search for the tip of the finger's shadow from an entire image to a line, as seen in Figure A.2. Because the shadow of the finger is created by the absence of the projector light, the shadow and the actual finger are related by the fundamental matrix.

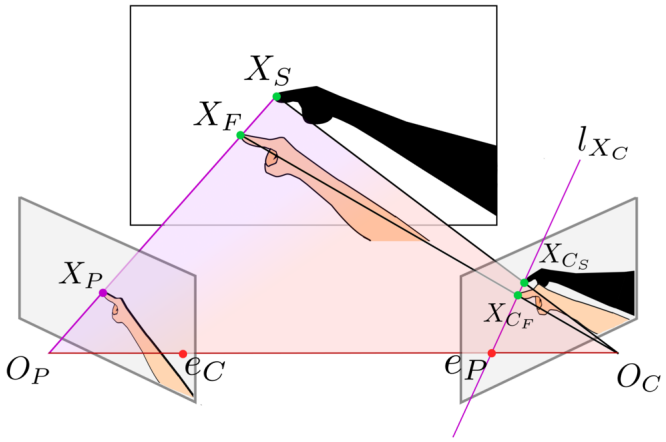


Figure A.2: Because a shadow is formed by the ray of light from the projector (here, centered at O_P), the image of tip of the finger X_{C_F} and its shadow X_{C_S} must lie on a line collinear with O_C.

REFERENCES

(2011). OLED Info. `http://www.oled-info.com/pentile`.

(2014). Coursera.

(2014). edX.

(2014). MIT OpenCourseWare.

(2014). Open Yale courses.

(2014). The SLIC browsing system.

(2014). UC Berkeley Extension Online.

(2014). Udacity.

Alvarez, C., R. Alarcon, and M. Nussbaum (2011). Implementing collaborative learning activities in the classroom supported by one-to-one mobile computing: A design-based process. *Journal of Systems and Software*, **84**(11), pp. 1961–1976.

Attewell, J. and C. Savill-Smith (2003). Learning with mobile devices: research and development. *mLearn 2003 book of papers*.

Banks, D. A. (2006). *Audience response systems in higher education: Applications and cases*. IGI Global.

Boatright-Horowitz, S. L. (2009). Useful pedagogies or financial hardships? Interactive response technology (Clickers) in the large college classroom. *International Journal of Teaching and Learning in Higher Education*, **21**(3), pp. 295–298.

Caldwell, J. E. (2007). Clickers in the large classroom: current research and best-practice tips. *CBE-Life Sciences Education*, **6**(1), pp. 9–20.

Chang, C. M.-T. . H. S., B. (2013). Understanding Students Competition Preference in Multiple-Mice Supported Classroom. *Educational Technology & Society*, (1), pp. 171–182.

Chen, X., J. Zheng, Y. Chen, M. Zhao, and C. J. Xue (2012). Quality-retaining OLED dynamic voltage scaling for video streaming applications on mobile devices. In *Design Automation Conference, 2012 49th ACM/EDAC/IEEE*, pp. 1000–1005. IEEE.

Cheung, N., D. Chen, V. Chandrasekhar, S. Tsai, G. Takacs, S. Halawa, and B. Girod (2010). Restoration of Out-of-focus Lecture Video by Automatic Slide Matching.

Chuang, J., D. Weiskopf, and T. Möller (2009). Energy aware color sets. In *Computer Graphics Forum*, volume 28, pp. 203–211. Wiley Online Library.

Cross, A., E. Cutrell, and W. Thies (2012). Low-cost audience polling using computer vision. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 45–54. ACM.

Cruz-Neira, C., D. J. Sandin, and T. A. DeFanti (1993). Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp. 135–142. ACM.

Davis, J. and X. Chen (2002). Lumipoint: Multi-user laser-based interaction on large tiled displays. *Displays*, **23**(5), pp. 205–211.

DeBourgh, G. A. (2008). Use of classroom clickers to promote acquisition of advanced reasoning skills. *Nurse Education in Practice*, **8**(2), pp. 76–87.

Dong, M., Y. Choi, and L. Zhong (????). Power modeling of graphical user interfaces on OLED displays. In *Design Automation Conference. DAC'09. 46th ACM/IEEE*, pp. 652–657.

Dong, M., Y. Choi, and L. Zhong (2009). Power-saving color transformation of mobile graphical user interfaces on OLED-based displays. In *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design (ISLPED'09)*, pp. 339–342.

Duan, L.-T., B. Guo, Y. Shen, Y. Wang, and W. L. Zhang (2013). Energy analysis and prediction for applications on smartphones. *Journal of Systems Architecture*.

Dyson, L., A. Litchfield, E. Lawrence, R. Raban, and P. Leijdekkers (2009). Advancing the m-learning research agenda for active, experiential learning: Four case studies. *Australasian Journal of Educational Technology*, **25**(2), pp. 250–267.

Falaki, H., R. Govindan, and D. Estrin (2009). Smart screen management on mobile phones.

Fan, Q., A. Amir, K. Barnard, R. Swaminathan, and A. Efrat (2007). Temporal Modeling of Slide Change in Presentation Videos. volume 1, pp. I–989–I–992.

Fan, Q., K. Barnard, A. Amir, and A. Efrat (????). Accurate alignment of presentation slides with educational video. In *Multimedia and Expo, ICME 2009*, pp. 1198–1201. IEEE.

Fan, Q., K. Barnard, A. Amir, A. Efrat, and M. Lin (2006). Matching slides to presentation videos using SIFT and scene background matching. pp. 239–248. ISBN 1-59593-495-2. doi:http://doi.acm.org/10.1145/1178677.1178710.

Fies, C. and J. Marshall (2006). Classroom response systems: A review of the literature. *Journal of Science Education and Technology*, **15**(1), pp. 101–109.

Fischler, M. A. and R. C. Bolles (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, **24**(6), pp. 381–395. ISSN 0001-0782. doi:http://doi.acm.org/10.1145/358669.358692.

Flinn, J. and M. Satyanarayanan (2004). Managing battery lifetime with energy-aware adaptation. *ACM Transactions on Computer Systems (TOCS)*, **22**(2), pp. 137–179.

Friedland, G., R. Rojas, and E. Tapia (2004). Teaching With an Intelligent Electronic Chalkboard. In *In Proceedings of ACM Multimedia 2004, Workshop on Effective Telepresence*, pp. 16–23.

Gigonzac, G., F. Pitie, and A. Kokaram (2008). Electronic slide matching and enhancement of a lecture video. In *Visual Media Production, 2007. IETCVMP. 4th European Conference on*, pp. 1–7. IET.

Graham, C. R., T. R. Tripp, L. Seawright, and G. Joeckel (2007). Empowering or compelling reluctant participators using audience response systems. *Active Learning in Higher Education*, **8**(3), pp. 233–258.

Han, K., Z. Fang, P. Diefenbaugh, R. Forand, R. Iyer, and D. Newell (2009). Using checksum to reduce power consumption of display systems for low-motion content. In *Computer Design, 2009. ICCD 2009. IEEE International Conference on*, pp. 47–53. IEEE.

Harter, T., S. Vroegindeweij, E. Geelhoed, M. Manahan, and P. Ranganathan (2004). Energy-aware user interfaces: an evaluation of user acceptance. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 199–206.

Hartley, R. and A. Zisserman (2003). *Multiple view geometry in computer vision*. Cambridge university press.

Iyer, S., L. Luo, R. Mayo, and P. Ranganathan (2003). Energy-adaptive display system designs for future mobile environments. In *MobiSys*, pp. 245–258.

Kay, R. H. and A. LeSage (2009). Examining the benefits and challenges of using audience response systems: A review of the literature. *Computers & Education*, **53**(3), pp. 819–827.

Kehl, R. and L. Van Gool (2004). Real-time pointing gesture recognition for an immersive environment. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pp. 577–582. IEEE.

Kennedy, M., H. Venkataraman, and G.-M. Muntean (2011). Energy Consumption Analysis and Adaptive Energy Saving Solutions for Mobile Device Applications. In *Green IT: Technologies and Applications*, pp. 173–189. Springer.

Kreitmayer, S., Y. Rogers, R. Laney, and S. Peake (2013). UniPad: orchestrating collaborative activities through shared tablets and an integrated wall display. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pp. 801–810. ACM.

Lin, C.-Y., F.-G. Wu, T.-H. Chen, Y.-J. Wu, K. Huang, C.-P. Liu, and S.-Y. Chou (2011). Using interface design with low-cost interactive whiteboard technology to enhance learning for children. In *Universal Access in Human-Computer Interaction. Applications and Services*, pp. 558–566. Springer.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, **60**(2), pp. 91–110. ISSN 0920-5691. doi:http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

Mahesri, A. and V. Vardhan (2005). Power consumption breakdown on a modern laptop. *Power-Aware Computer Systems*, pp. 165–180.

Maynes-Aminzade, D., R. Pausch, and S. Seitz (2002). Techniques for interactive audience participation. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, p. 15. IEEE Computer Society.

Mundy, D., D. Stephens, and K. Dykes (2010). Facilitating Low Cost Interaction in the Classroom Through Standard Mobile Devices. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, volume 2010, pp. 1819–1825.

Nickel, K. and R. Stiefelhagen (2003). Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. In *Proceedings of the 5th international conference on Multimodal interfaces*, pp. 140–146. ACM.

Oh, J.-Y. and W. Stuerzlinger (2002). Laser pointers as collaborative pointing devices. In *Graphics Interface*, volume 2002, pp. 141–149. Citeseer.

Park, S., Y. Lee, J. Lee, and H. Shin (2005). Quality-adaptive requantization for low-energy MPEG-4 video decoding in mobile devices. *Consumer Electronics, IEEE Transactions on*, **51**(3), pp. 999–1005.

Pylyshyn, Z. W. and R. W. Storm (1988). Tracking multiple independent targets: Evidence for a parallel tracking mechanism*. *Spatial vision*, **3**(3), pp. 179–197.

Smith, A. (2010). Mobile access 2010. *Washington, DC: Pew Internet & American Life Project*.

Stowell, J. R. and J. M. Nelson (2007). Benefits of electronic audience response systems on student participation, learning, and emotion. *Teaching of psychology*, **34**(4), pp. 253–258.

Swaminathan, R., M. E. Thompson, S. Fong, A. Efrat, A. Amir, and K. Barnard (2010). Improving and Aligning Speech with Presentation Slides. *Int. Conf. Pattern Recognition (ICPR) 2010*, pp. 3280–3283.

Thornton, P. and C. Houser (????). Using Mobile Phones in Education. In *WMTE '04: Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'04)*. ISBN 0-7695-1989-X.

Tung, Q., M. Korp, C. Gniady, A. Efrat, and K. Barnard (????). MobiSLIC: Content-aware Energy Saving for Educational Videos on Mobile Devices.

Tung, Q., R. Swaminathan, A. Efrat, and K. Barnard (2011). Expanding the point—automatic enlargement of presentation video elements. In *Assocation of Computing Machinery Multimedia*, ACM MM.

Vogt, F., J. Wong, S. Fels, and D. Cavens (2003). Tracking multiple laser pointers for large screen interaction. In *Extended Abstracts of ACM UIST*, pp. 95–96.

Wang, F., C.-W. Ngo, and T.-C. Pong (2003). Synchronization of lecture videos and electronic slides by video text analysis. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pp. 315–318. ACM, New York, NY, USA. ISBN 1-58113-722-2. doi:http://doi.acm.org/10.1145/957013.957080.

Wang, F., C.-W. Ngo, and T.-C. Pong (2007). Lecture video enhancement and editing by integrating posture, gesture, and text. *Multimedia, IEEE Transactions on*, **9**(2), pp. 397–409.

Wang, X. and M. Kankanhalli (2009). Robust Alignment of Presentation Videos with Slides. In *Advances in Multimedia Information Processing-PCM 2009: 10th Pacific Rim Conference on Multimedia, Bangkok, Thailand, December 15-18, 2009. Proceedings*, pp. 311–322. ISBN 978-3-642-10466-4. doi:http://dx.doi.org/10.1007/978-3-642-10467-1_27.

Ware, C. (2012). *Information visualization: perception for design*. Elsevier.

Wee, T. K. and R. K. Balan (2012). Adaptive display power management for OLED displays. In *Proceedings of the first ACM international workshop on Mobile gaming*, pp. 25–30. ACM.

Winslow, A., Q. Tung, Q. Fan, J. Torkkola, R. Swaminathan, K. Barnard, A. Amir, A. Efrat, and C. Gniady (2009). Studying on the move: enriched presentation video for mobile devices. In *2nd IEEE Workshop on Mobile Video Delivery (MoViD), in conjunction with INFOCOM*.

Zdravkovska, N., M. Cech, P. Beygo, and B. Kackley (2010). Laser Pointers: Low-cost, Low-tech Innovative, Interactive Instruction Tool. *The Journal of Academic Librarianship*, **36**(5), pp. 440–444.

Zualkernan, I. A. (2011). Infocoral: Open-source hardware for low-cost, high-density concurrent simple response ubiquitous systems. In *Advanced Learning Technologies (ICALT), 2011 11th IEEE International Conference on*, pp. 638–639. IEEE.