EduCase: An Automated Lecture Video Recording, Post-Processing, and Viewing System that Utilizes Multimodal Inputs to Provide a Dynamic Student Experience

by

Sara T. Itani

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013

Author .....
Department of Electrical Engineering and Computer Science
August 8, 2013

Certified by .....
Frédo Durand
Professor
Thesis Supervisor

Accepted by ....
Dennis M. Freeman
Chairman, Department Committee on Graduate Theses

EduCase: An Automated Lecture Video Recording, Post-Processing, and Viewing System that Utilizes Multimodal Inputs to Provide a Dynamic Student Experience

by

Sara T. Itani

Submitted to the Department of Electrical Engineering and Computer Science
on August 8, 2013, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

This thesis describes the design, implementation, and evaluation of EduCase: an inexpensive automated lecture video recording, post-processing, and viewing system. The EduCase recording system consists of three devices, one per lecture hall board. Each recording device records color, depth, skeletal, and audio inputs. The Post-Processor automatically processes the recordings to produce an output file usable by the Viewer, which provides a more dynamic student experience than traditional video playback systems. In particular, it allows students to flip back to view a previous board while the lecture continues to play in the background. It also allows students to toggle the professors's visibility in and out to see the board they might be blocking. The system was successfully evaluated in blackboard-heavy lectures at MIT and Harvard. We hope that EduCase will be the quickest, most inexpensive, and student-friendly lecture capture system, and contribute to our overarching goal of education for all.

Thesis Supervisor: Frédo Durand
Title: Professor

Acknowledgments

I would like to express my sincere gratitude to the following people for their support throughout the duration of this project.

- Professor Frédo Durand for agreeing to be my supervisor despite the perhaps unusual introduction.[1] It was a pleasure working with you this past year. Your guidance and positive attitude kept me motivated throughout the duration of this project.
- Edwin Guarin and the rest of the folks at Microsoft for ~~fueling my Kinect-obsession~~ supporting me with resources and technical guidance.
- Brandon Marumatsu (OEIT) and James Donald (edX) for providing resources, feedback, and encouragement throughout the duration of the project.
- Riccardo Campari, Fang-Yu Liu, and Adin Schmahmann for your code contributions to EduCase. It was fun working with you all.
- John Romanishin for the gorgeous 3D-printed Surface-tripod mount.
- Professor Robert Winters, Professor Babak Ayazifar, and Professor Randall Davis for allowing us to field-test the recording system in your classes.
- All the chess-players at Harvard Square for helping me procrastinate. This project wouldn't have been nearly as ~~stressful~~ exciting without you :).
- Jonathan Poggi for being a true friend all these years (and helping me proofread at the very last minute.)
- Last, but not least, I would like to thank my loving family. Mom, Dad, Nadia, and Adam - thank you for providing feedback and for being so supportive throughout this project and life in general.

[1] I'm running around CSAIL attempting to find a thesis supervisor. I ask a friend whether she knows any professors interested in online education initiatives, and she mentions Professor Frédo Durand. Two minutes later, I'm in his office. He's in a meeting. Oops. He's a bit confused about who I am and why there. "I'm looking for a thesis supervisor. When would be a good time to drop by totally unannounced and pitch an idea for an MEng project." "How about tomorrow, 2pm?"

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Universities all around the world are posting their course materials online as part of the OpenCourseWare (OCW) and Massive Online Open Course (MOOC) initiatives. MIT in particular has been an active participant, having spearheaded the effort and posted over two thousand courses thus far. MIT OCW offerings include materials ranging from syllabi to assignments to lecture notes. Of all the offerings, students find lecture videos the most useful and engaging: as of November 2011, each course with a video lecture had 18 times more engagement than the average OCW course. [4]

But despite the high demand, only 2% of MIT OCW videos have lecture videos attached to them.[1] This is because recording video is extremely expensive: it requires costly recording equipment, editing software, and countless man hours. By some estimates, it costs up to $800 to record/edit a single hour of lecture, which makes it unlikely for video content to be produced for more inconspicuous courses.[2]

Even with such a high price tag, the remote learning experience remains subpar. Whereas a student sitting in a lecture hall can simply glance back at a previous board, a student watching a lecture video must pause the video, rewind to the relevant

[1] Of MIT's 2,083 published courses, only 48 are full video lectures. [4]

[2] Estimate from an informal chat with Professor V.A. Shiva, who had once expressed interest in recording his lectures. The high cost could be related to union wages.

Figure 1-1: Students glancing in different directions during lecture (some at the professor, some at the board, some at a previous board, and some at their notes.) Current lecture videos cannot emulate this dynamic and personal experience.

location, and fast-forward when they are ready to move on to the rest of the lecture. A cameraman, no matter how experienced, simply cannot emulate the dynamic and personal experience of sitting in a lecture hall (Figure 1-1).

Furthermore, there are even features of a lecture that are bothersome to students sitting in the lecture hall itself. For instance, sometimes the professor inadvertently blocks the blackboard they are referencing (Figure 1-2).[3] Unfortunately, even when the professor is mindful to quickly move away from the board after writing something down, the recorded lecture suffers because the cameraman tends to follow the professor around the room instead of focusing on the most relevant board. This makes it difficult for students to fully absorb the material without interrupting their workflow.

[3]Video that exemplifies many of the problems described above: http://www.youtube.com/watch?v=k6U-i4gXkLM

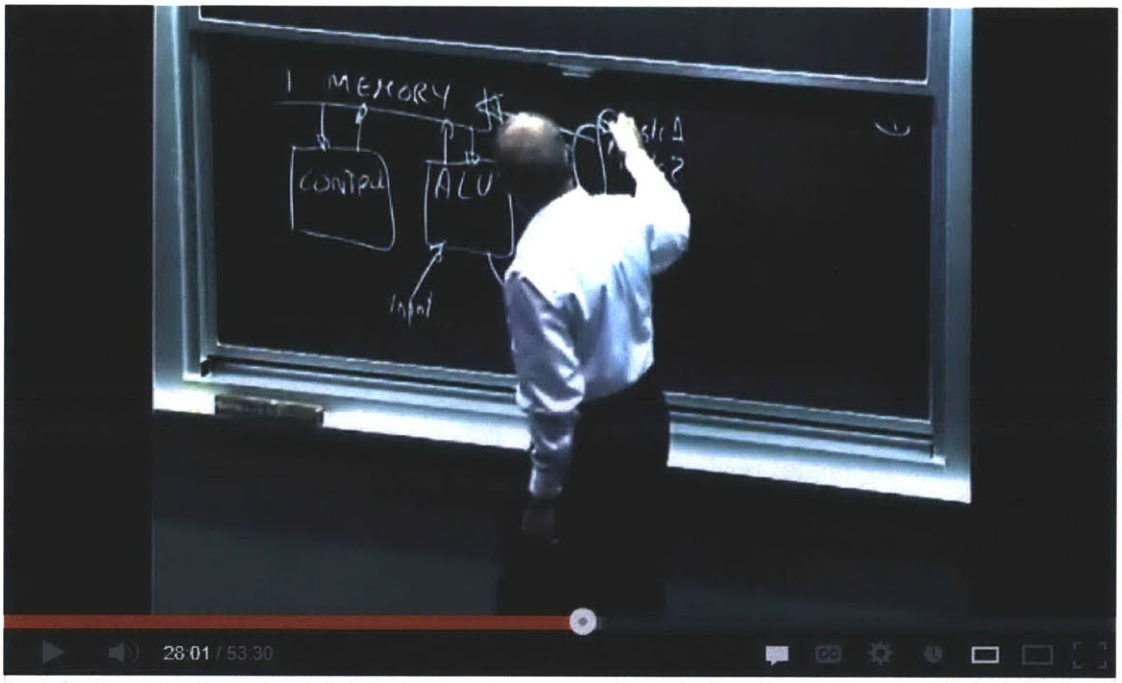


Figure 1-2: Sometimes, professors inadvertently block the board they are trying to explain.

1.2 Vision

EduCase is an inexpensive and portable automated lecture video recording, editing, and viewing system that aims to both reduce the cost of producing content, and enable a more dynamic and personal student experience.

A professor simply walks in, unfolds his EduCase Recorder, and presses a button for a hassle-free lecture recording experience. When the lecture is over, the content is automatically post-processed, and posted online for students to view within twenty-four hours.

Researchers have invested a lot of time and effort on automating the lecture recording/editing process, but much of it has focused on emulating the cameraman, which unfortunately results in a subpar student experience as described in Section 1.1. EduCase, on the other hand, puts students first by enabling them to customize every element of their experience so that it is comparable, or even more satisfying, than being physically present in a lecture hall. They will be able to play back the lecture at any speed they choose, toggle the professor's visibility to view obstructed material, change the viewing angle, rewind or skip through material as they choose, and even obtain lecture summaries so they can be sure they didn't miss anything.

The EduCase recording system consists of three devices (one per lecture hall board.) Whereas a traditional video camera only records color and audio inputs, each EduCase recording device records color, depth, skeletal, and audio inputs. Once the recording is complete, we automatically post-process the data to produce a more dynamic student experience.

In particular, EduCase provides the following features:

- **Prioritizes student experience:** As described in Section 1.1, a single cameraman cannot capture the dynamic nature of a lecture. Therefore, EduCase employs multiple cameras so that students may interact with the entire scene rather than a small subset. This enables students to multi-task in the same way they do during a live lecture (e.g. viewing a previous board while the lecture continues to play in the background). Furthermore, EduCase uses a depth

camera to filter out the professor from the image. Students can "look behind" the professor to view information on the board the professor may be blocking.

- **Intelligent system** EduCase will automatically focuses on the correct board by using the Microsoft Kinect to track the professor, detect gestures (e.g. writing or pointing to the board gesture,) and do voice recognition.

- **Scalable:** Because EduCase will use multiple smaller cameras to capture the view rather than a single high-quality camera with complex panning/zooming features... the system will be relatively portable and inexpensive.

- **Expandable to non-traditional lecture settings:** because there's a trend away from the traditional lecture-hall format, the system will be designed so that it is easy to add functionality by simply recording new gestures

- **Inexpensive, portable, and unobtrusive recording system**

1.3 Contributions

This thesis describes the preliminary design, architecture, implementation, and evaluation of EduCase an automated video lecture recording, and viewing system. The key contributions of this thesis are:

- We develop an inexpensive, portable, and unobtrusive set of recording devices that records audio, color, depth, and skeletal streams in a format usable by the EduCase Post-Processor.

- We develop a Post-Processor that combines the inputs from multiple EduCase Recorders that to automatically apply various effects to prepare the lecture for students. It utilizes gestures to determine when to switch views, it utilizes the depth and color streams to create a video that allows students to see through the professor in case he is blocking the board, and it synchronizes the audio streams.

- We develop a lecture viewing application that provides students with a more dynamic experience than present-day lecture video playback systems. In particular, we enable students to toggle the professor in and out of view, and view a previous board while the lecture continues to play in the background.

Although further work is required to realize the vision outlined in Section 1.2, we believe EduCase has potential to be the quickest, easiest, most-inexpensive way to record, post-process, and view video lectures.

1.4 Thesis Overview

This thesis is organized as follows. Section 2 provides a brief survey of related works and describes how EduCase builds upon them. Section 3 describes the design and architecture of EduCase. It introduces the preliminary research, ensuing design goals, and the resulting approach. The design and architecture of the Recording, Post-Procesing, and Viewing modules are discussed, along with challenges encountered during implementation. Section 4 ties everything together by walking through the end-to-end user experience. Section 5 describes the feedback received during field-testing at MIT and Harvard. Lastly, Section 6 discusses the future direction of the project, followed by a conclusion of the thesis.

Chapter 2

Related Work

This chapter provides a brief survey of current and in-development video lecture recording, editing, and viewing technologies, and describes how EduCase differentiates itself and contributes.

2.1 One-to-One Cameraman Emulation

Many automated lecture recording systems have sought to emulate the cameraman, complete with various audio/video effects that serve to make videos more engaging, such as panning back and forth or zooming in and out. Nagai's system [10] does online processing of the lecture to control a Pan-Tilt-Zoom camera. However, such online methodologies can result in imperfect shots. Virtual Videography [9] addresses this problem by using simple syntactic cues to post-process fixed camera footage to produce lecture videos similar to those composed by a cameraman.

One-to-one cameraman emulation, however, is non-ideal for students. Students are often multi-tasking during lecture: they are looking at the board, referring back to previous boards, watching the professor, and taking notes. A single cameraman, however, can only focus on one thing at a time. So whereas a student sitting in lecture can simply glance back at a previous board, students watching video lectures must pause the video, rewind back to a previous frame, fast-forward to their original position, and continue playback.

[9] cites Bordwell [6] in its claims that cinematic effects are necessary because "a camera in a fixed location do not provide enough visual interest or guidance to the viewer." However, we believe such effects do not provide as significant an impact in lecture videos as they do for films: the shots in films are significantly more complex than those in a lecture hall, and cinematic effects are not necessary to direct attention to relevant material. The lecturer's use of gestures and speech is sufficent to direct the students's attention to key material in an otherwise static environment.[1]

In fact, videographers recording lectures are actually encouraged to use multi-camera setups to achieve different shots instead of panning and zooming to avoid distracting students trying to focus on what's written on the board. [5]

As described in a review of a lecture video series, some students feel that the camera work often detracts from an otherwise "well planned and delivered" lecture:[1]

> **Good Instructor... fire the cameraman**
>
> Well planned and delivered lectures. The instructor presents the material in a clear and logical progression of ideas leading up to the main point. The camera work on the other hand is annoying and distracting. It suffers from too much movement, constant zooming, panning, and tilting, in an attempt to follow the instructor or reframe the writing on the board. This should be the easiest job in the world. Frame the picture on the blackboard and lock the camera off. The important information is on the board. Following the instructor around the room adds nothing to the subject, and takes my attention away from the information on the board. I don't need to see the instructor's lips move to hear what he is saying. Unfortunately, so many other filmed lectures suffer the same problem. Why these idiot camera people can't be taught to keep their hands off the camera is beyond me.

Similar to [10] and [9], EduCase automatically post-processes the recordings to make it easy for students to focus on the important material. However, we forgo the

[1] "It is well known that humans are extremely sensitive to the gaze attention of others." [8]

panning/zooming effects in favor of a customizable student experience of the viewing experience.

2.2 Distributed Recording Systems

In addition to the problematic issues raised in Section 2.1, high-fidelity cameraman-emulation is relatively inflexible. This limits the classrooms in which lectures may be digitally captured.

Microsoft Research (MSR) takes an interesting approach to the problem. [11] Rather than relying on a single camera, they make use of multiple cameras, each capturing a different perspective a couple cameras are pointed at the lecturer, and one camera is pointed at slides. The camera pointed at the slide automatically detects slide changes, and therefore displays the view of that camera when appropriate. MSR's system has proven to be effective after having undergone several years of field-testing.

This multi-camera approach allows more flexibility with regards to camera placement because no one camera need be capable of capturing the entire scene. Similarly, EduCase utilizes multiple recording devices, each capturing a different view. However, whereas MSR focuses on PowerPoint presentations, EduCase applies the concept to university lectures with a whiteboard/chalkboard component where it is more challenging to decide on the relevant information to display. Furthermore EduCase benefits from increased portability and simpler setup process.

2.3 Dynamic Viewing Experiences

ClassX is an inexpensive lecture capture system that provides a customizable student experience by allowing students to pan and zoom around the classroom themselves. [2] While ClassX automatic scene selection works best for PowerPoint presentations, however too much interaction is required on the part of the student in the case of blackboard-heavy lectures. EduCase differentiates itself from ClassX by putting more

emphasis on automatic scene selection in blackboard-heavy environments.

It has been demonstrated in Kinect-enabled dressing rooms that multimodal inputs can be used to provide a more dynamic experience. [3] While such systems do not focus on lecture capture, they serve to inform and inspire our work with EduCase.

2.4 Summary

EduCase builds on the work described in the sections above. EduCase aims to strike a balance between Virtual Videography and ClassX - the automation of Virtual Videography with the customization of ClassX. We do this by employing a distributed recording device setup similar to MSR's, which allows us to record the entire scene in a way that makes syntactic sense and allow students to select different scenes as they see fit. EduCase uses multimodal inputs similarly to Kinect-Enabed-Fitting-Rooms to make better cinematic decisions as well as provide a more dynamic student experience. Ultimately EduCase differentiates itself by being both cost effective and providing a compelling student experience (Figure 2-1).

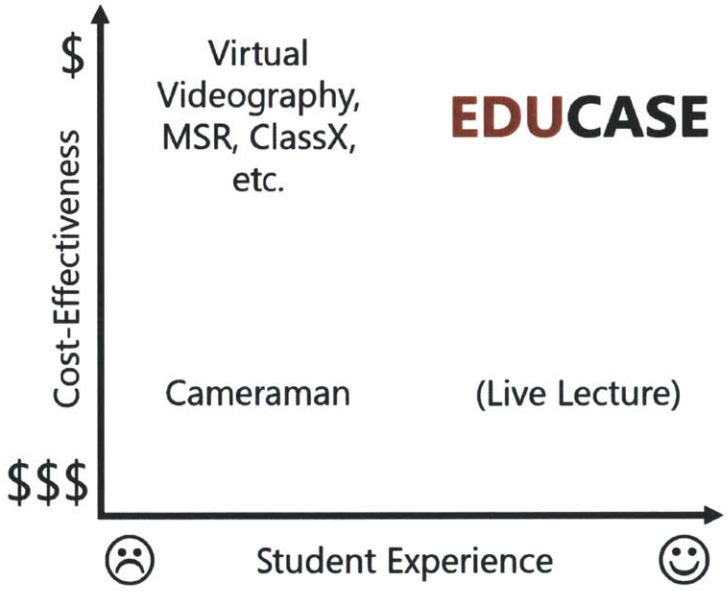


Figure 2-1: Comparison of EduCase to other Lecture Capture/Post-Processing systems

Chapter 3

Design and Architecture

3.1 Preliminary Research

Before embarking on the project, we spoke with James Donald from edX, a MOOC platform, about what video processing techniques they use before they post video lectures online, as well as what sort of student feedback they've received. This helped us decide what we should focus on with regards to both video editing techniques and the potential integration of our system with the EdX framework.

From the meeting, we learned that student satisfaction does not increase significantly with video quality so far as the writing on the board is legible (Figure 3-1). Audio quality, on the other hand, is extremely important.

Therefore, we decided not to prioritize high resolution video or fancy panning/zooming effects as in previous research when building EduCase. We did, however, design the system to be integrate with standard lapel microphones.

Along the same line, it is critical that the audio is closely synchronized with the video because it may distract viewers from the video as the audio-video offset increases over fifty milliseconds. [7]

Furthermore, one video processing technique EdX uses (but isn't immediately obvious,) is editing out silences (e.g. after a professor asks a question to the class) and/or audience chatter in the video. Such voids are much more distracting during a lecture video than during a live lecture.

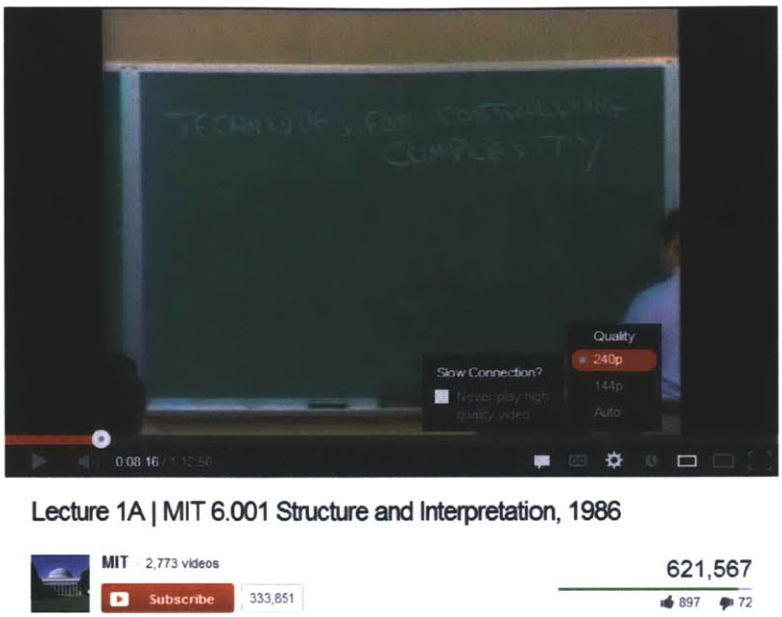


Figure 3-1: Video quality is unnecessary for high student satisfaction, as demonstrated by the favorable reviews on a 240p video of an MIT OCW lecture.

Lastly, one of the biggest student complaints about lecture videos was having to pause and rewind the video in order to reference a previous board.

3.2 Design Goals

It is important to have clear goals in mind when designing such a large system so that it becomes easier to make tradeoffs and limit the scope of the project. This section explores the design goals of EduCase that were established based on the preliminary research described in Section 3.1.

3.2.1 User Experience Goals

We express our user experience goals in terms of the "stories" listed below. This method helps us keep the end-goals of our users in mind as we design the system.

- **Simple Setup:** As a professor I don't want to worry about getting to lecture early to set up EduCase. I want it to "just work" in whatever environment I'm

faced with.

- **Unobtrusive Setup:** As a student/professor, I do not want to be distracted during lecture.

- **Lecture Recording:** As a professor, I want to be able to record my lectures so I can post it online for my students to learn from.

- **Automated Lecture Editing:** As a student/professor want the lectures to be engaging, correct, helpful, and processed in a timely manner for the students who are watching them.

  Auto-Select Boards: As a student, I want the camera to always focus on the most relevant information to me with little or no work on my part.

  Dynamic Environment: As a student I may be focusing on multiple things at the same time and I want to be able to be able to "look around the room" at whatever content I might be interested in - even if it is different from the content that the majority of my peers are interested in seeing at any particular time.

  Invisibility Cloak: As a student, I want to see behind the professor while he is writing on the board. At the same time, I don't want interaction to disappear completely from the picture. I want to know where the professor is, and what he is pointing to. As a professor, I want to enable this experience without feeling useless.

  Ease of Understanding: As a student, I want to be able to effortlessly understand the audio and video content.

- **Infrastructure:** As a student/professor, I want a seamless end-to-end experience without having to worry about connecting things together myself

3.2.2 Framework Goals

We prioritize the following software design principles:

- **Simplicity/maintainability:** We favor readability over performance unless the performance gain is necessary and/or significant. This is especially important since our goal is to pass the code on to EdX at some point. Adherence to this principle also simplifies division of labor.

- **Modularity/scalability:** In order to account for the wide range of lecture hall setups and student preferences, we favor a modular design that makes it easy to add additional cameras, post-processing steps, and features.

- **Graceful degradation:** According to Murphy's law, "Anything that can go wrong, will go wrong." We acknowledge that our post-processing routine will not be perfect. Therefore, in the event of failure, it is important that the application experience gracefully degrades, so that students do not miss out on important lecture material.

3.2.3 Prototype Goals

Finally, in accordance with the principles above, we have established the following goals and non-goals for the prototype described in this thesis.

We seek to develop a three-stage recording, post-processing, and viewing system that meets the goals as shown in Figure 3-2.

Recorder →	Post-Processor →	Viewer
Goals		
• Minimal Setup • Portability • Flexibility to be used in variety of lecture hall setups • Records audio, color, depth, skeletal inputs • Unobtrusive to students/professor in class • Inexpensive compared to current solutions • Battery life must last duration of lecture	• Minimal human interaction • Post-process input from recording devices to produce content usable by viewer • Post-processing input from recording devices to produce content usable by Viewer • Human-editable output	• Ability to view previous boards • Ability to compensate for errors in post-processor • Automatically select most relevant board • Toggle professor visibility in and out of view
Non-Goals		
• Live streaming during recording • High quality video	• Professional effects (e.g. panning back and forth, zooming in and out)	• PowerPoint presentation support: many existing lecture capture systems incorporate this feature, so we shall not focus on it for now. • Skipping through portions of video with silences/audience chatter: It has been demonstrated that this is possible, so we design with addition of this feature in mind, but do not require it.

Figure 3-2: Prototype goals and non-goals

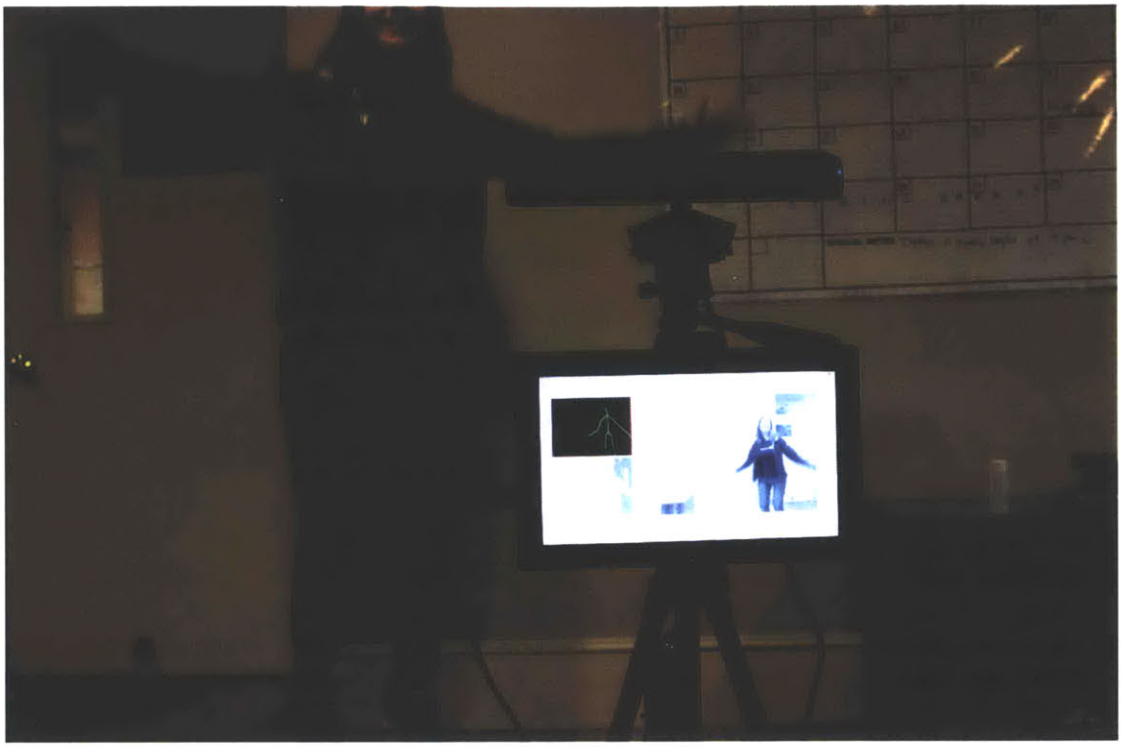


Figure 3-3: Each recorder records color, depth, skeletal, and audio streams.

3.3 Recorder

This section describes the design and implementation of the EduCase Recorder

3.3.1 Hardware

EduCase consists of three recording devices, one per lecture hall board (Figure 3-4 and Figure 3-5). A distributed recorder setup allows us to capture the full scene while simultaneously providing flexibility to be used in a variety of lecture hall setups. Furthermore, professors often segment lecture topics by board, so focusing on a single board provides a simple but effective semantic method of focusing on the most relevant content. Such a setup is also favored by videographers over panning/zooming effects as described in 2.1. This setup also allows us to choose more inexpensive hardware, which may have a small field of view or weak zooming capabilities.

Each device consists of a Microsoft Kinect sensor (Figure 3-6) and a PC (64GB Microsoft Surface Pro), which allows it to records four data streams: color, depth,

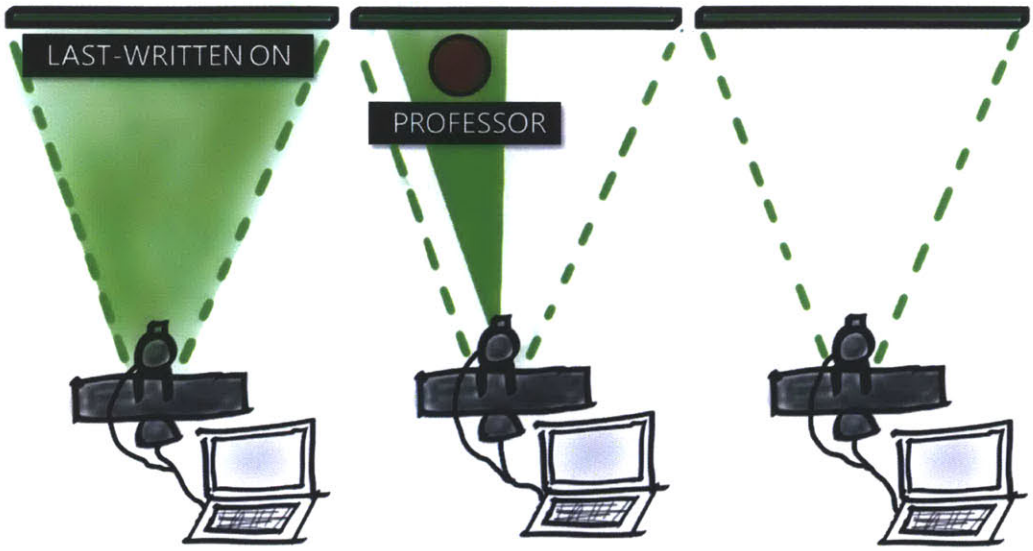


Figure 3-4: The EduCase recording system consists of three devices, one per lecture hall board. Each device has a Kinect and a PC.

skeletal, and audio. This allows us to easily track the professor and retrieve gestural information, which provides enough information to determine the most relevant board. We combine the depth and color streams to determine the precise position of the professor in the frame for use in the invisibility cloak post-processing module.

The Kinect interacts with the PC using the Kinect for Windows API. We chose Kinect for Windows over its open source counterparts, OpenNI and libfreenect, because it does predictive analysis. Therefore it is much better suited for non-optimal environments (e.g. a classroom) where it may lose track of a body part because it's behind something (e.g. a podium.)

The PC we chose for this application is the Microsoft Surface Pro because it satisfies the following requirements:

- Power: must be able to compress and save multiple streams of data without losing frames.

- Portability: the slim form factor and drop-friendly SSD make it transportation-

Figure 3-5: EduCase Recorder: Kinect, PC, battery pack, mounted on a tripod

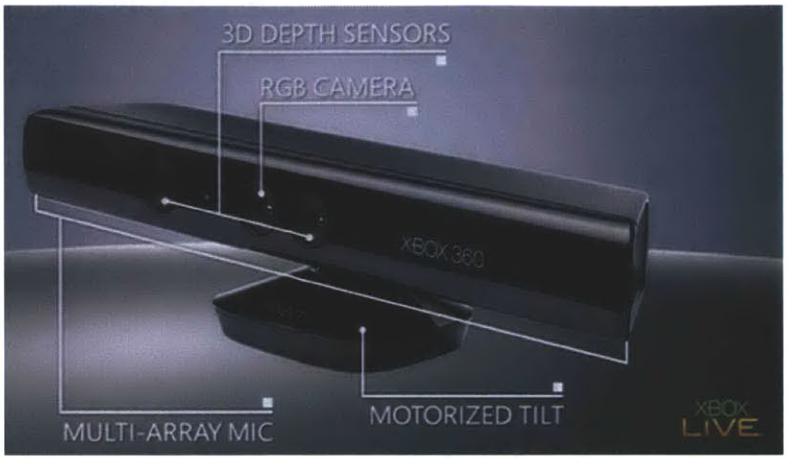


Figure 3-6: Kinect Sensor

friendly.

- Battery life: must last the duration of a 3 hour lecture

- Touchscreen: easier to control when mounted on a tripod

- USB 3.0: A USB port is also necessary in order to use the Kinect. Furthermore, the lecture recordings can take up several gigabytes per device, so USB 3.0 makes the file-transfer experience bearable.

The tripod keeps the Kinect mounted 2.5 feet high and placed 12 feet from the board Figure 3-8), which we empirically discovered was optimal for reducing noise when the professor is not facing the Kinect. This setup was also un-intrusive for both students and professors: it provides professors plenty of space to move around the classroom, and also allows students to sit immediately behind the device without obstructing their view of the board (Figure 3-7).

In order to make the device more portable and easy to set up, we mount the Kinect and Surface on a foldable tripod (Figure 3-9). We also modified the standard Kinect power cable to use rechargeable batteries instead of an outlet (Figure 3-10).

Figure 3-7: Recorder doesn't block board even in unslanted classrooms

Figure 3-8: Scene captured by Kinect when placed twelve feet away from board.

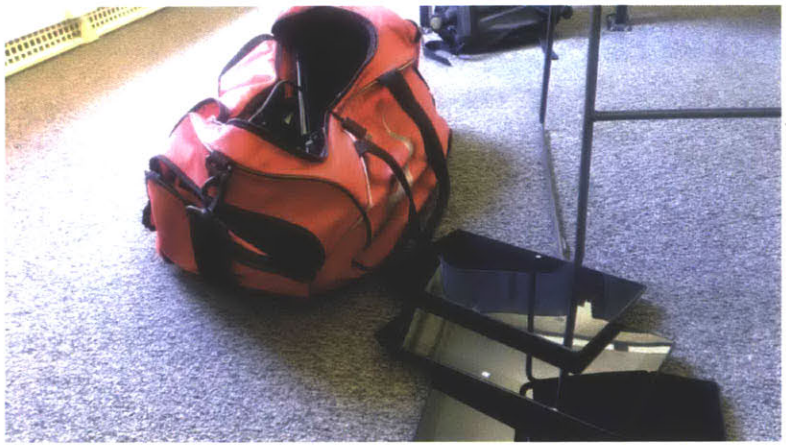


Figure 3-9: Portability: everything necessary for recording a three-board lecture fits inside a small duffle bag.

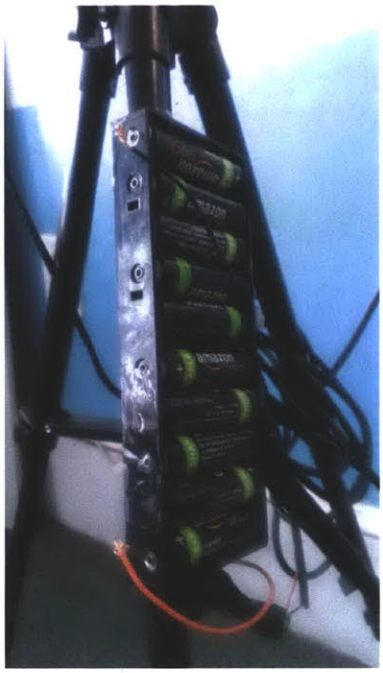


Figure 3-10: The standard Kinect power cable has been modified to accept power input from 10 rechargeable AA batteries instead of an AC outlet, which allows the device to be more portable.

3.3.2 Software

The EduRecorder is responsible for recording color, depth, skeletal, and audio stream data from the Kinect to disk.

We faced a significant memory/performance challenge because the of the large amount of data produced by the Kinect.

Storing the data in an uncompressed format would result in a 400 GB file for a 90 minute session, which would both take a huge toll on the hard-disk and decrease the reliability of the device.

We chose to compress the color and depth streams because they were contributed most significantly to file size. There's a plethora of color video codecs available, so compressing the color stream was simple enough. Compressing the depth stream was a two step process. First, each depth point cloud is converted to a color image as shown in Figure 3-11, so that different depths corresponding to different colors. We then save the depth stream to a video file, which allows us to use standard video

Figure 3-11: Compressed depth stream: darker colors correspond to points that lie further away. Bodies are signified by a hue value.

codecs instead of inventing our own compressions scheme.

Table 3.1 summarizes the files produced by the recorder during the course of a single lecture. We split up the files in this fashion to make it easier to seek through the compressed video during post-processing.

However, once we compressed the data, we began to run into performance issues. The compression process took a major toll on our CPU, so we kept losing frames (Figure 3-12).

We solved this performance problem by utilizing a lookup table to convert depths to color pixels in the histogram, and reducing the codec settings to favor speed over compression.

Initially, we were worried that our depth data would become too noisy since video compression can distort some of the colors. However, we've found this difference to be unproblematic for our purposes.

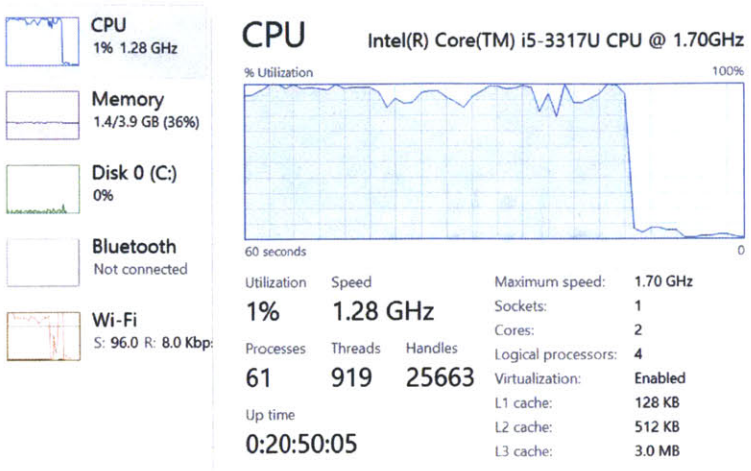


Figure 3-12: Overworked CPU before using lookup tables to compress depth streams (CPU usage drop-off occurs when recording ends.)

3.4 Post-Processor

The post-processing module takes as inputs three .eduCaseRecording files (one from each device,) as well as a high quality audio recording of the lecture.

The post-processor runs the files through the Gesture Engine and Invisibility Cloak modules in parallel to produce a post-processed video. All of the input devices are synchronized through associated audio stream.

The post-processor produces a .eduClient file that links to processed audio/video files as well as describes keypoints and synchronization offsets for use in the Viewer.

3.4.1 Gesture Engine

The Gesture Engine combines the skeletal and depth point cloud modalities in order to detect and recognize gestures (Figure 3-13). We use "important" gesture keypoints to automatically switch to the right board at the right time.

Gesture Recognition

Initially, we implemented a Hidden Markov Model to recognize gestures. However, we soon learned that even though we could recognize explicit gestures, we had no

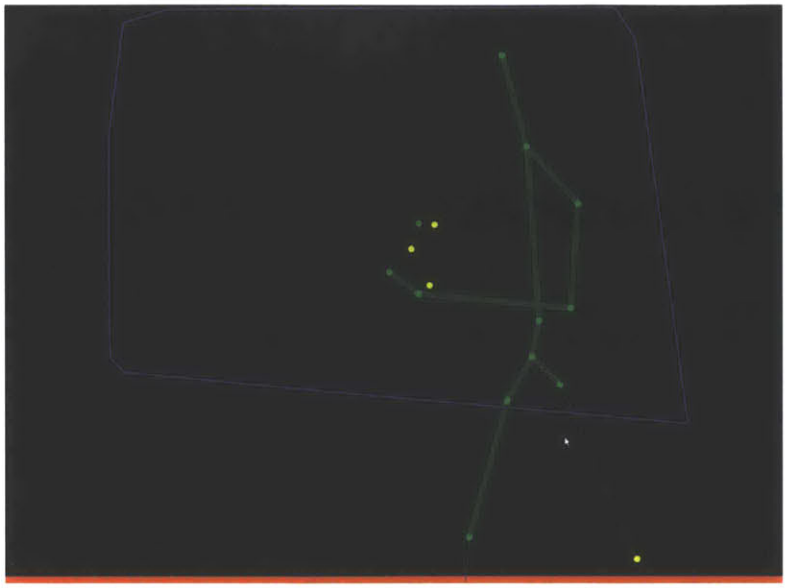


Figure 3-13: The Gesture Engine recognizes gestures based on skeletal input stream and blackboard location (purple) as specified by user during blackboard selection process.

robust way to detect the start of a gesture. Moreover, it required us to gather more data than we had time for.

We eventually settled on a simple but effective state machine approach that detects two gesture types: POINTING, BLACKBOARD_INTERACTION. These gesture types were chosen because they allow us to detect the last-interacted-with-board while ignoring boards the professor may just be pacing in front of.

This simple approach was sufficient to automatically detect the most relevant board. State machines often suffer from the fact that it is difficult to define a state machine that works well for every person and every gesture they could make. This issue is unproblematic for us, however, because even if the gesture recognition engine produces a false negative (professor begins interacting with the board, but this goes undetected,) it is likely that the professor will continue interacting with the board in some way, so the gesture is likely to be recognized soon enough.

One challenge we overcame was that there would often be multiple skeletons in view due to either students passing in front of the device or to noisy data. Instead of choosing between skeletons, we attached a gesture engine to every tracked skeleton,

Figure 3-14: The Kinect uses IR light to detect depth. This causes depth data from blackboards appears noisy because black objects absorb infrared light.

so that a gesture recognized event would be raised for any skeleton in view.

For now, we have found the BLACKBOARD_INTERACTION and POINTING gestures we've defined to be sufficient for deciding on which board is most relevant. However we plan to improve on/add to them in the future. For instance, distinguishing between writing and erasing gestures may provide us with a better semantic understanding of the lecture, which could allow us to provide a better student experience.

Detecting Blackboard Location

One problem with blackboards is that they are... well... black. Black objects tend to absorb infrared light, which is problematic because the Kinect uses an infrared light projection in order to detect depths. Figure 3-14 shows how the depth data is incomplete around the blackboard.

Therefore, we must determine the location of a blackboard without relying on depth data directly from the blackboard. The blackboard selection process (3-15 prompts the user to select points on the wall around the blackboard, establishing that a blackboard is the plane defined by the convex hull of points.

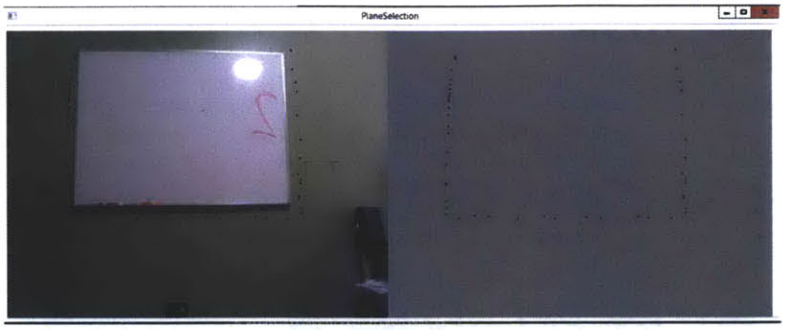


Figure 3-15: User selects points around blackboard as part of the blackboard selection process.

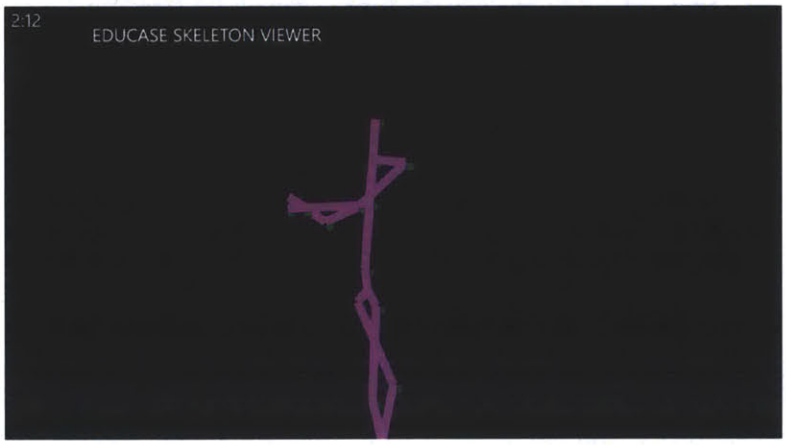


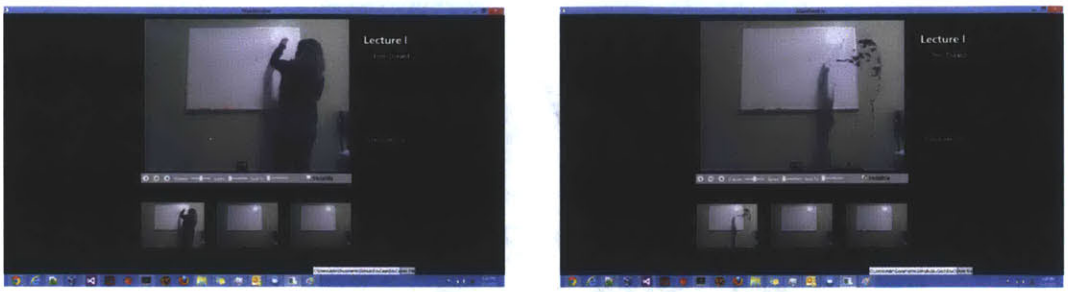
Figure 3-16: Debugging tangled skeleton using phone application that streams live data from Kinect sensor.

Debugging Gestures

It was very time consuming to record/test/debug individual gestures. Therefore, we built a mobile application and framework that made this easier to debug and iterate by interfacing with the Kinect. The application enables us to see the Kinect skeleton on a phone. Additionally, it includes buttons to begin/stop/label gesture recordings. This was especially helpful since a lot of our gestures involve facing the board (instead of a computer screen.) Figure 3-16 shows us debugging a tangled skeleton position.

3.4.2 Invisibility Cloak

It's always frustrating when professors stand in front of the material they are trying to explain. I've sat in many a class wishing there was some sort of button to just



(a) (b)

Figure 3-17: Invisibility cloak allows students to see board behind professor.

make them disappear...

The invisibility cloak combines the depth point cloud and color streams in order to allow students to see the board the professor is blocking.

The depth point cloud includes information about whether there is a body present at each pixel. As soon as a there is no body present at a pixel for several frames (threshold determined empirically,) we add that pixel to the background.

When post-processing, we actually move backwards through the video so that the background is actually built from the future board (rather than the previous board) to handle cases when the professor is standing in front of material he is currently writing on the board.

When the student toggles on the invisibility cloak, we filter out any visible bodies, and display a the background in their place.

We faced several challenges in building the invisibility cloak. As you can see from Figure 3-17b, the Kinect occasionally has difficulty identifying the edges of bodies or hair as part of a skeleton, which causes them to end up in the background (and remain visible when the invisibility cloak is on). We reduce the effect of this noise by effect by waiting until the surrounding pixels have not been occupied by a person for a certain period of time before updating the background. We ultimately decided removing the noisy data completely wasn't worth the performance loss because the process must be completed for every frame.

Another issue is that the infrared and color cameras on the Kinect are slightly offset, so there is not necessarily depth information available for every color pixel

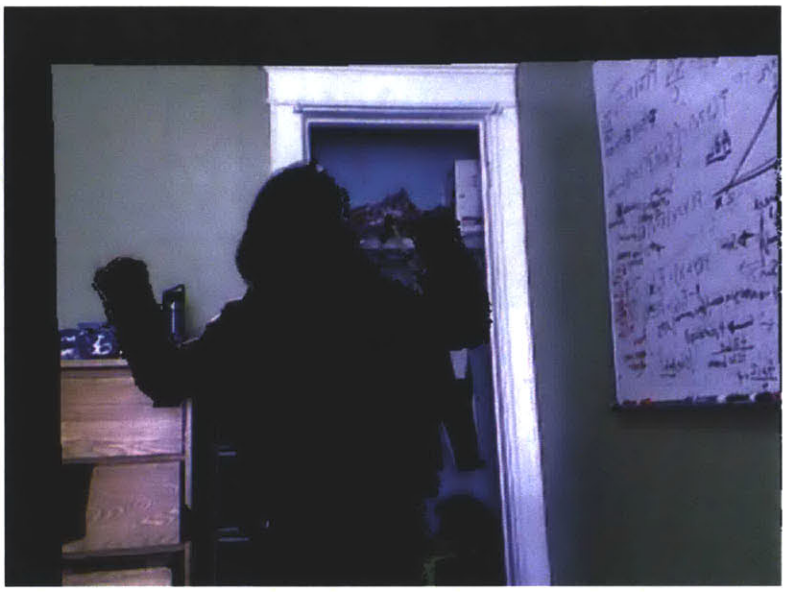


Figure 3-18: Edges of invisibility cloak prone to error due to offset between depth camera and color camera on Kinect.

(Figure 3-18). This means that the edges are prone to error.

3.4.3 Synchronizing Multiple Modalities

One challenge we faced was synchronizing the disparate recording devices. The one commonality between all of these devices is that they all have microphones. So we chose to synchronize all of the inputs based on the audio stream. In order to do this, we use normalized cross correlation to find the time shift between audio signals. Synchronizing everything with audio also satisfies one of our design goals of high quality audio by making it easy to incorporate high quality microphones.

3.4.4 Human-Editable Output

The output files from the post-processor are read by the Viewer. In accordance with the graceful degradation principle described in Section 3.2.2, the output files of the Post-Processor are XML formatted so that they are human-readable in case of error.

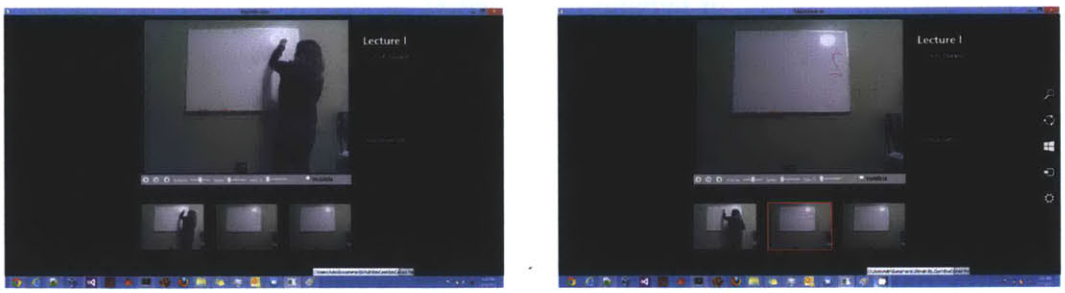
3.5 Viewer

The EduCase Viewer brings together the post-processed recordings from each of the three recorders to provide an enhanced student experience compared to traditional lecture videos.

Because we are using a fairly simple automation approach, we are certain that we will make some mistakes. This is why we designed the Viewer to provide a graceful degradation into a manual mode so that the student can control which board is being watched by simply clicking on one of the thumbnails below the primary video (Figure 3-19).

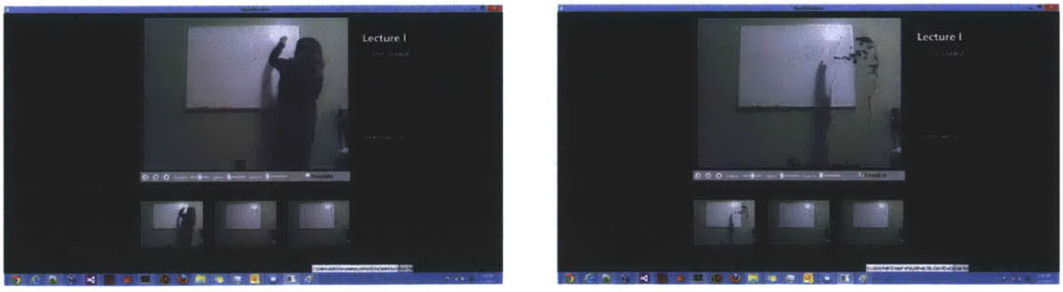
It turns out that in addition to making up for post-processing errors, this actually provides a better experience than OCW videos recorded by a cameraman. A common complaint regarding OCW videos is that the cameraman never seems to be able to focus on the right thing. And how can we blame him? Lecture is traditionally a very dynamic experience. A student might be watching the board, watching the professor, or watching an old board. A cameraman can't possibly focus on everything at the same time. With EduCase, however, we try our best to automatically focus on the right board, but we also make the entire scene available to students, so that a student can simply flip back to a previous board while the lecture continues to play in the background. Taking a cue from chat applications like Google Hangouts and Skype, we maintain an outline around the manually-selected view.

Students can toggle on/off the invisibility cloak so that they can see the board behind the professor (Figure 3-20) In the invisibility cloak mode, an outline is maintained around the professor so that students know where the professor is gesturing.



(a) (b)

Figure 3-19: Instead of relying on the board automatically selected by EduCase, students can simply click to view a previous board while the lecture continues to play in the background. Clicking on the manually selected board once more returns the application to auto-board-selection mode.



(a) (b)

Figure 3-20: Viewer allows students to toggle invisibility cloak on and off.

File Suffix/Extension	Description
.eduCase	compilation of all frames, with links to depth, color, skeleton frames in external files. This is a small file, so we can load the entire thing into memory, and seek through it. Kinect parameters, information about the recording, pointers to other files, links to frames in color/depth recordings
-color.avi	x264-based encoding of color stream
-depth.avi	x264-based encoding of depth stream (after point cloud is converted to color image)
.skel	binary format skeletal frames
.wav	audio stream recording from Kinect

Table 3.1: Files recorded by each recording device

Chapter 4

Walkthrough

This section describes the end-to-end experience enabled by the EduCase Recorder, Editor, and Viewer in order to get a better sense of how everything ties together.

4.1 Recorder

5-10 minutes before class starts, the TA sets up the three EduCase Recorders so that one device is pointing at each board. Recording begins as soon as each device is situated.

The professor wears a lapel microphone, and begins recording as soon as he is ready to start lecture. The recording devices can last up to three hours on battery.

When lecture is over, the professor stops the recording on all of the devices, packs up the devices, and leaves.

4.2 Post-Processor

A USB flash drive is used to transfer the recordings from each of the devices (the three EduCase Recorders, and the lapel microphone) to a more powerful computer for post-processing.

The post-processing application prompts him to select the chalkboard in the three camera recordings, after which the professor leaves the computer for several hours as

it completes processing.

4.3 Viewer

The students browse to the course they want to view, and plays it back.

The application does a good job of automatically selecting the board the student is most interested in, so the interaction required on the part of the student is minimal.

When a student needs to glance back to a previous board (or the application wrongly selects a board), he/she simply clicks back to view a previous board while the lecture continues to play in the background.

If the professor is standing in front of a board the student wants to see, the student can simply toggle on the invisibility cloak until the professor is no longer in obstructing their view. An outline is visible so that the students can still see what the professor is pointing at.

Traditional video playback features (pause, play, change speed, fast-forward, rewind, etc.) are also available.

Chapter 5

Evaluation

As shown in Figure 5-3 and Figure 5-2, we have begun testing EduCase in MIT's 6.041 (Probabalistic Systems Analysis), 6.853 (Intelligent Multimodal User Interfaces), and Harvard's S-21b (Linear Algebra). All of these courses are relatively chalkboard-heavy, which makes them ideal for EduCase. This field-testing has enabled us to iterate on the performance of the system itself, as well as highlighted some interesting elements of lectures that we shall take into account while building future versions of EduCase.

5.1 Post-Processing Performance

"The first law of computing is that it hates you." Adin Schmahmann

We'd tested everything many times over, and we were excited to start taking the device around with us to class. Naturally, the first time we attempted to record a real lecture, we were horrified to see a memory leak rear its ugly head (Figure 5-1). Eventually, we managed to fix the bug, and record a few sessions.

Overall, the system performed very well, and managed to select the most relevant board within a few seconds of the professor interacting with it. It also performed fairly well in the face of high podiums and tables in between the students and the professor.

Name	Status	72% CPU	91% Memory	51% Disk	0% Network
Recorder		71.4%	2,719.4 MB	4.0 MB/s	0 Mbps
Windows Explorer		0%	53.6 MB	0 MB/s	0 Mbps

Figure 5-1: Our first field test came to an abrupt halt when this memory leak reared its ugly head.

Lecture halls with too much sunlight proved problematic because it interferes with the infrared sensors on the Kinect. This was easily solved by closing the window shades near the chalkboards, but we acknowledge that there are some lecture halls that may not have simple a fix.

Sometimes a student would inadvertently bump into the device on their way into lecture. In these cases, we did a good enough job of repositioning the device such that the recording was not ruined. However, future iterations of the recorder should detect when it's been moved and prompt the user to reselect the chalkboard in the post-processing stage.

5.2 Professor/Student Reception

Professors were excited by the prospect of recording their lectures without needing significant technical capability or dealing with hassle of hiring a cameraman to record the class. Students were excited by the invisibility cloak and the ability to switch between views of different boards while the lecture continues to play in the background.

One of our goals was for the recording system to be unobtrusive to students and professors alike. Therefore, we took special care to observe their behavior with the introduction of the device.

At first, some students were distracted by the device when the screen was on. This was swiftly rectified by turning the screen off during recording.

It has been demonstrated many times over that people tend to change their behavior when they feel they are being watched,[1] so we were worried about the obtrusiveness of the EduCase recorders which require placement at the very front of the room. We

[1]This phenomenon is known as the Hawthorne Effect.

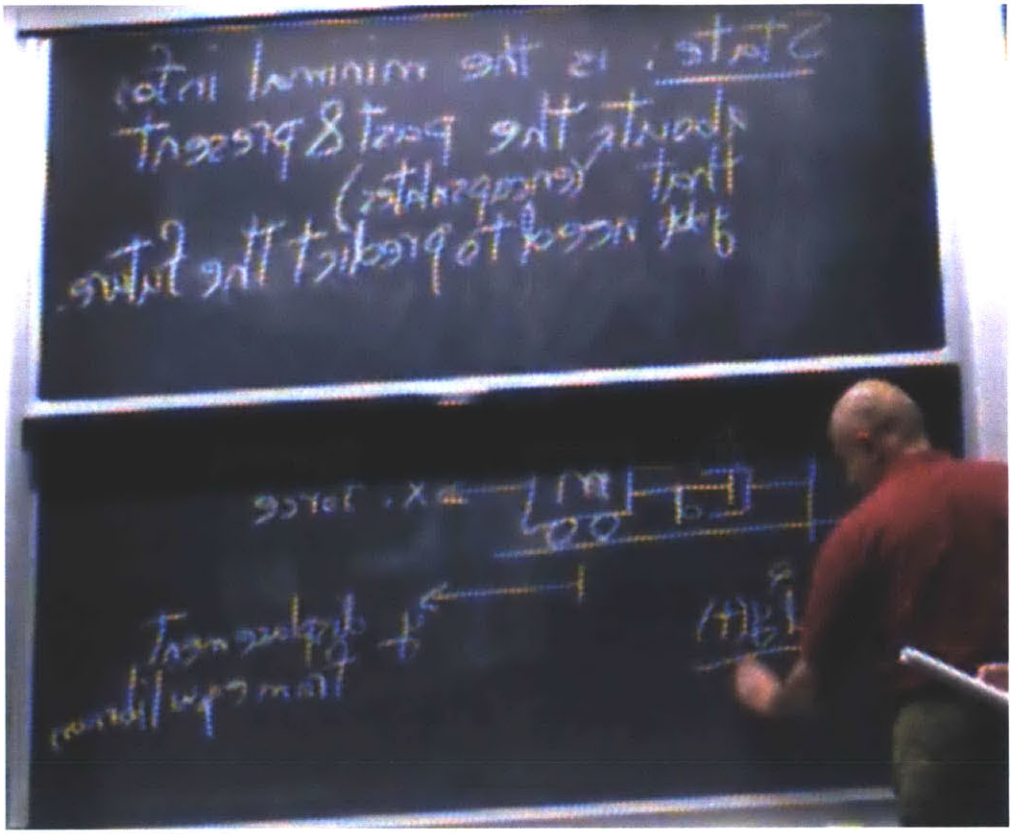


Figure 5-2: View from Recorder while testing EduCase in MIT's 6.041

Figure 5-3: Testing EduCase in Harvard's S-21b.

did not want professors to subconsciously or consciously alter their teaching style, and we also did not want students present in the class to feel uncomfortable or have their views blocked. Initiatives such as Virtual Videography intentionally placed cameras at the very back of the room in order to avoid this phenomenon.

It turns out that this was not an issue for us in practice. Neither the professors nor the students seemed to exhibit any changes in behavior. After recording a session of Math 21-b at Harvard Summer School, Professor Winters mentioned that he felt absolutely no pressure from the device. He mentioned that the only behavior change he noticed was that there was actually a certain point in lecture where he stopped and wondered why he *wasn't* changing his style in face of the cameras. Winters hypothesizes that it may be because the design of the device doesn't make it look like a camera that is "aggressively pointing towards towards their subject". This makes sense: the Kinect was designed for placement in a living room: if millions of people are comfortable keeping a Kinect in their own living rooms, it makes sense that people would not find it particularly intrusive in a classroom setting.

As for the students, we initially noticed some hesitation from the first-row-dwellers, who were worried they might interfering with the recording if they walked in front of the device. However, as soon as the first row had a few people in it, students became comfortable passing in front.

There was only one case in which the device "disrupted" a student. 20 minutes into lecture, a girl walks in late, walks straight to the front row and center seat where one of the recording devices was sitting, and without hesitation, moves the device aside, and sits in the seat. I walk to the front of the room and reposition the device. Initially we didn't think anything of it - the screen was off, so maybe she just didn't realize the device was on and recording. However, next time - as if on cue - this same girl walks in 20 minutes late, and without hesitation, moves the device aside, and sits in *her* seat. Our initial threat model wrongly assumed that students would not interfere with the device. We therefore suggest the addition of the following feature: an electric shock should be induced whenever a recording device is touched during a lecture. An alternative, more insurance-friendly, option is to simply inform students

that they are not supposed to move the device while it is recording.

Chapter 6

Conclusion

6.1 Summary of Contributions

In this work, we presented EduCase: an inexpensive automated lecture recording, post-processing, and viewing system that provides a more dynamic student experience than traditional lecture videos. We evaluated EduCase in classrooms at MIT and Harvard.

In Chapter 1, we described and motivated the goal of a better automated lecture video capture system. In Chapter 2, we described how EduCase builds on a body of related works. We described how EduCase differentiates itself by providing a more dynamic student experience than traditional lecture videos.

Chapter 3 described the main contributions of this thesis. In particular, we first established a set of design goals based on some preliminary research. We then detailed the design and architecture of the Recording, Post-Processing, and Viewing modules. The currently implemented functionality of each of these modules is summarized below.

- Recorder: inexpensive and simple setup set of recording devices

  Physical device consisting of a tripod, Microsoft Kinect, Microsoft Surface, modified Kinect power cable (to add battery pack), 3D-printed Surface-tripod holder.

Application that records color, depth, skeletal, and audio streams

- Post-Processor: produces processed files in format usable by Viewer with minimal human interaction

  Gesture engine to detect pointing/writing on board gestures

  Mobile application to make it easier to debug and record sample gestures

  Invisibility cloak to remove a professor from view and display future board in place

  Synchronization of recording devices based on audio stream

  Human-editable output file

- Viewer: custom lecture viewing client that provides a more dynamic experience

  Automatic board selection based on gesture engine

  Ability to toggle between normal and invisible professor mode

  Ability to view a previous board while the lecture continues to play in the background

Chapter 4 provided a better sense of how the system fits together as a cohesive story.

Chapter 5 detailed the evaluation of EduCase. In particular, we described the reasonable performance of the post-processor, as well as potential improvements. We also described the relatively good reception of the recording devices in classrooms, and concluded that they were unobtrusive.

6.2 Future Work

6.2.1 Improvements in Output Quality

We chose not to prioritize video quality for the current iteration of EduCase. Currently, lecture recordings are illegible when professors write using small text. There-

fore we plan to incorporate higher resolution video, either by mapping the Kinect with an external HD camera or by upgrading to the recently announced Kinect 2.

Furthermore, we hope to expand EduCase to encompass PowerPoint lectures.

Lastly, we plan to crowdsource data from students watching the videos online to improve the quality of our post-processing. For instance, if a statistically significant number of students switch to Board 1 even though our automated approach focuses on Board 2, the system will eventually learn to switch to Board 1 instead. Along the same line of thought, instead of monopolizing the lecture editing techniques ourselves, it might be better to create an open platform for developers to enhance the student experience by providing them access to the gestures, audio keypoints, etc.

6.2.2 New Application Areas

In addition to simply editing the video, the gestural and/or audio information from the Kinect recorders could be used to enhance the student experience in other ways. For instance, distinguishing between WRITING and ERASING gestures would allow us to create a compilation of boards in their "completed" states. This series of boards could serve as lecture notes. But even more than that, each of these boards is associated with a particular point in the video, so they could provide an enhanced navigation experience that allows students to semantically zoom in/out of the lecture.

A distributed gesture-based automated capture system also has applications outside of academia. Gestural information could be used to gain a simple semantic understanding of a sports game, so it could be used to automatically select and switch to the most relevant camera.

6.3 Parting Thoughts

EduCase has had a warm reception thus far. It recently won 3rd place in MIT's iCampusPrize 2013. We are also speaking with edX and MIT OEIT about further development of the system. There are plenty of issues to be worked out before EduCase can actually be deployed, but it is exciting to see everything finally start to

come together.

We anticipate that lectures will eventually evolve from their traditional formats, but this will likely take many years as it would require a significant transformation in teaching style. EduCase will make it simple and inexpensive to record any currently offered course in a student-friendly manner, which will enable a large corpus of lecture material to be posted online without the need to develop new content.

Bibliography

[1] Calculus i - download free content from florida international university on iTunes. https://itunes.apple.com/us/itunes-u/calculus-i/id397848271?mt=10. Accessed 2013-08-07.

[2] ClassX: home. http://classx.stanford.edu/ClassX/. Accessed 2013-08-07.

[3] Kinect virtual dressing room at topshop lets ladies 'Try on' clothes. http://www.huffingtonpost.com/2011/05/11/kinect-dressing-room_n_860740.html. Accessed 2013-08-07.

[4] Mit ocw — november 2011 dashboard report. http://ocw.mit.edu/about/site-statistics/monthly-reports/MITOCW_DB_2011_11.pdf. Accessed 2013-08-07.

[5] Recording video | technology solutions for teaching and research. https://academictech.doit.wisc.edu/?q=node/32. Accessed 2013-08-07.

[6] David Bordwell. *On the History of Film Style.* Harvard University Press, January 1998.

[7] Scott Firestone, Thiya Ramalingam, and Steve Fry. *Voice and video conferencing fundamentals.* Cisco Press, first edition, 2007.

[8] Marianne Gullberg and Sotaro Kita. Attention to speech-accompanying gestures: Eye movements and information uptake. *Journal of Nonverbal Behavior*, 33(4):251–277, 2009.

[9] Rachel Heck, Michael Wallick, and Michael Gleicher. Virtual videography. *ACM Transactions on Multimedia Computing Communications and Applications (TOMCCAP)*, 3(1):4, 2007.

[10] Takayuki Nagai. Automated lecture recording system with AVCHD camcorder and microserver. In *Proceedings of the 37th annual ACM SIGUCCS fall conference*, SIGUCCS '09, pages 47–54, New York, NY, USA, 2009. ACM.

[11] Cha Zhang, Yong Rui, Jim Crawford, and Li-Wei He. An automated end-to-end lecture capture and broadcasting system. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4(1):6:1–6:23, February 2008.