First Experiences with a Classroom Recording System

Paul E. Dickson
Hampshire College
School of Cognitive Science
893 West St.
Amherst, MA 01002, USA
pdickson@hampshire.edu

W. Richards Adrion, Allen R. Hanson,
and David T. Arbour
Department of Computer Science
Computer Science Building
University of Massachusetts
Amherst, MA 01003, USA
{adrion,hanson,darbour}@cs.umass.edu

ABSTRACT
This paper describes our experiences with the first partial deployment of Presentations Automatically Organized from Lectures (PAOL), a lecture recording system developed and tested at the University of Massachusetts Amherst. PAOL automatically records all information presented during lectures using any combination of computer, whiteboard, and overhead presentation and compiles the captured lectures into indexed presentations. We discuss lessons learned from this deployment that have application in lecture recording specifically and classroom technology in general. We also discuss our initial evaluation of created presentations as determined by a small focus group study.

Categories and Subject Descriptors
H.5.1 [Multimedia Information Systems]: Video; I.4.1 [Digitization and Image Capture]: Sampling; I.2.10 [Vision and Scene Understanding]: Video analysis; K.3.0 [Computers and Education]: General

General Terms
Algorithms, Design

Keywords
Automatic Capture, Lecture Recording, Image Processing, Video Creation, Indexing, Evaluation

1. INTRODUCTION
The prevalence and cost of digital cameras, wireless microphones, and technology in general have made it possible to shift lecture recording from a fiscal impossibility to a real possibility for universities and colleges. Research [1, 9] has shown that giving students access to recorded lectures has a positive effect on learning. While many lecture recording systems exist [2, 5, 6, 8, 9, 10, 11], none can record whiteboard, computer, and overhead projection material while remaining transparent to the lecturer (discussed below). For this reason we developed Presentations Automatically Organized from Lectures (PAOL) [4]. In this paper we describe the lessons learned from a semester’s worth of partial deployment of PAOL as well as initial student reaction to the recorded presentations.

PAOL is a cost-effective lecture capture system that was developed and tested on lectures captured in the Computer Science department of the University of Massachusetts Amherst. PAOL not only records a lecture but also organizes the material recorded from any combination of computer, whiteboard, and overhead as an indexed presentation (Figure 1). This presentation gives students far better access to the content than that provided by a simple video of the lecture. The content delivery system is written in Flash, giving students access to recorded lectures from any device with a modern Web browser.

Figure 1: PAOL presentation GUI.

Our initial deployment demonstrated the effectiveness of PAOL to create presentations that students find useful. We also learned that PAOL will need to be modified before full adoption can be implemented. Some of the lessons learned apply not only to PAOL but in general to all lecture recording systems and classroom technology. We will begin the paper by describing how PAOL relates to other lecture recording systems before giving a brief overview of how PAOL works in order to put our results in context. We will then describe our initial deployment and testing of PAOL and discuss what we learned. We will finish with a discussion of redesigns underway and proposed.

2. RELATED WORK
Many lecture capture systems have been created and deployed at many colleges, but unfortunately none of these
Authoring on the Fly [5] and Streams [8] are both automatic lecture recording systems that focus primarily on recording a video of the lecturer and provide navigation primarily through timeline sliders. Microsoft has a similar lecture recording system [6] that can automatically index material but only if PowerPoint is used. A system at Berkeley [7] functions primarily as a Web-casting system with video and slides, again limiting lecturer presentation options.

The 2 most automatic and dynamic systems are MediaSite [10] and TeleTeachingTool (TTT) [11]. MediaSite provides automatically indexed presentations and can index any material presented using a computer but requires a user-supplied lecturer video and does not capture whiteboard material. TTT can index whiteboard and computer material but does so by requiring an electronic whiteboard and all software to be run on a computer running the TTT software. PAOL has none of the restrictions of either of these systems.

PAOL automatically and dynamically captures computer, whiteboard, and overhead projection and automatically creates presentations that index all 3 types of material [4]. It uses an external screen capture device to capture computer material and cameras to capture whiteboard and overhead material, making the recording process transparent to the lecturer. The image processing techniques used on each of the captured data streams identify changes in the presented material, enabling PAOL to index any material and not just that presented in a specific manner, such as lecture slides. The only requirement that PAOL places upon lecturers is that they must wear a wireless microphone.

3. SYSTEM DESIGN

PAOL uses a pair of Flea2 cameras from Point Grey Research and a VGA2USB capture device from Epiphan to capture all presented content. These capture devices are connected to a processing server (Figure 2). By using the VGA2USB capture device, we enable the lecturer to use any computer without any restriction placed on the software used. The camera views provide high resolution (1024x410 pixel) images of whiteboard material that we process for clarity as well as enable us to create a dynamic lecturer video that tracks the lecturer. The cameras allow us to capture both without requiring either a pan-tilt-zoom camera to track the lecturer or an electronic whiteboard.

Figure 1 shows how PAOL presentations consist of whiteboard content, computer content, a video of the lecturer, and an index. Except for the index, each of these content fields is created by a separate algorithm, the highlights of which are described below and in detail in [3, 4]. The index generation does not require a separate algorithm as it is created by temporally integrating the whiteboard and computer-generated index points.

3.1 Whiteboard Capture

The goal of the whiteboard capture algorithm is to identify, clarify, and save content presented on the whiteboard. The process used to clarify the text is described briefly here and in detail elsewhere [4]. The basic concept is that images captured by the cameras are processed to first whiten the whiteboard in order to identify text before having the lecturer removed so that no text is occluded. The text is clarified through a process of increasing the image contrast and sharpening, and the image is compared with images captured before and after to identify when changes have occurred. In Figure 3, a sample before and after image processing shows the system’s robustness under poor lighting conditions. When changes occur, an index point is generated and the image is saved; sample saved images can be seen in Figure 4. As can be seen in Figure 5, saving images where changes have occurred allows the progression of ideas to be recorded. Because the process uses images of the front of the room, any material, including that from an overhead projector, is captured in the same manner (Figure 4).

3.2 Lecturer Video

The lecturer video is created by extracting small lecturer-centered video frames from the images used to capture the whiteboard and animating these frames together to create a
video that shows the lecturer location in relation to current material being presented. The algorithm looks for differences between consecutive captured images and then identifies the lecturer location based on these differences (Figure 6). This location is then used to crop the lecturer video frame. Details on how this works and the smoothing techniques used to improve video quality were published elsewhere [4].

3.3 Computer Capture

The computer capture algorithm, similarly to the whiteboard capture algorithm, looks for content changes to save. It waits for material being presented to stabilize before saving the content; sample captured images can be found in Figure 7. Stabilization ensures that all content presented is captured but that quick slide transitions (when a lecturer jumps from one part of a lecture to another) and mid-annotation tablet PC changes are ignored (Figure 8).

3.4 Presentation GUI

The Flash-based presentation GUI can be seen in Figure 1. All windows can be resized to enlarge one or more of content windows. For greater legibility the images in the thumbnail-based GUI darken when the corresponding section of the video is playing and enlarge when the cursor is scrolled over them.

4. TESTING

The algorithms used to capture all of the content have been discussed in detail with statistical breakdowns of results ([3, 4]) but until the Fall 2008 semester PAOL had not been deployed and used by students during the course of a semester. Our deployment consisted of installing PAOL in a classroom and attempting to record 2 separate courses for the entire semester. Our goal was to evaluate the effectiveness of PAOL both as a class recording mechanism and a teaching and learning technology. As to the former, we hoped to begin our evaluation by testing the number of

Figure 4: Sample captured whiteboard and overhead images.

Figure 5: Progression of whiteboard material.

Figure 6: Top: The original image; left: cropped lecturer video frame; and right: lecturer location identified.

Figure 7: Sample computer captured images.

Figure 8: Sample cropped whiteboard and computer captured images.
whiteboard index points, number of computer index points, timing of both, quality of captured data, effectiveness of the GUI, and general quality of the system overall. To evaluate its impact on learning, we designed an assessment strategy based on surveys and interviews.

4.1 Deployment

We chose 2 courses from a number of faculty volunteers. Each met twice a week on different days in order to allow enough processing time between recordings. Our intention was to record all meetings of each class and to get the recorded presentations up within 24 hours of the lecture. This did not occur and the reasons for this proved quite instructive.

The first issue discovered related to the process of scheduling the lecture recording. Our goal was transparency, so the system had to be able to run on a given schedule without the lecturer starting the capture software. PAOL was developed in Windows because we believed that its general user friendliness and the large number of multimedia applications for it would make it the desired platform for a deployable system. We chose Windows over Macintosh because of larger market share and general lower cost of equipment. For several reasons, automated scheduling proved problematic. First, the scheduler required that the process owner be logged in at the scheduled start time. Second, Windows has no native scripting language. Our final solution to the scheduling problem involved 5 separate programs. For reasons that we could never determine, our capture setup failed to run or quit part way through a lecture during 52.6% of all lectures. We concluded that it is more important to use an operating system built around a multiple user setup that can run processes in the background and that has built-in scripting and scheduling than to use one that is more user friendly.

The scripting itself was also part of the problem. Windows is designed to primarily allow a user to access all information via point and click. This means that software used under Windows generally cannot be run from a command line; the scripting language must be able to effectively point and click its way through applications in order to run them. Since the options selected are related to the window in focus on the screen and Windows allows programs to create pop-up windows that make them the focus, the scripts written would occasionally fail because the wrong program was in focus. Additionally, the scripting language would sometimes focus on the incorrect window within an application. These were not the usual cause of the failures. Again, this problem can be solved by using a command-line-based operating system that includes scripting.

The next issue was related to the wireless microphone we used. Part way through the semester, a professor in the department who was teaching at the same time began recording the audio of his lectures using the same wireless microphone. Since the microphones only have 2 frequencies, both microphones occasionally operated on the same frequency, destroying PAOL’s audio track. From this we learned that it is very important to keep everyone informed of recording plans.

4.2 Focus Groups

We developed questionnaires for students in both classes and planned follow-up interviews with some. The recording problems interfered with this plan. In addition, the students were not able to easily download or stream the lectures because the full presentations averaged around 400MB and the streaming video option did not work well off our campus network. Most of the students did review at least one lecture, but this did not give them a sufficient picture of PAOL presentations from which we could get meaningful statistical data.

For these reasons we shifted to a focus group approach to collecting data so that we could ask specific questions and gauge answers based on student familiarity with the presentation. The focus groups consisted of both scheduled one-on-one interviews with 5 students that lasted between 45 minutes and 1 hour and covered all areas of the PAOL presentations. We also gathered anecdotal evidence from discussion with larger portions of the class.

4.3 Results

From discussions with students, we learned that they were only likely to review the presentations if they were consistently recorded and easy to download. If the students could not quickly and easily get them the first time they tried, they were unlikely to try again. That said, all students responded positively to the idea of the presentations and looked forward to having access to them more consistently in the future.

Most importantly, all students expressed interest in the system and asked whether it would be possible to have PAOL created presentations available for their next class as they thought it would be useful in all of their classes. One student noted that having a PAOL presentation consistently during the semester would have made it far easier for him to review the material he missed during one lecture that he was too sick to attend.

We also learned that the quality of the lecturer video does not need to be extremely good as long as it is not the primary method of content delivery. Students noted that the video had some jumps and jitter, caused by unresolved hardware issues, but stated that this was not a problem. The general consensus was that they liked being able to see the lecturer but aside from that the quality did not matter as they were focusing on the computer content window; they were glad the video window was there but no improvement in quality was necessary.

A preliminary study (Dickson, Adrion, Hanson, unpublished, 2008) showed that the quality of captured content was unimportant if there were navigation and interface issues in the GUI, which led to improvements in the GUI. This new version of the GUI is what is described above. Students had no problem with the thumbnail-based index.

Students also noted that all computer, whiteboard, and overhead content appeared to occur at the correct times in relation to the lecturer video and that all content presented
appeared to have been captured, but they said that since they had not been using the presentations all semester, they could not state this conclusively. All preferred the processed versions of the whiteboard captured content.

We also interviewed the professors teaching the 2 courses and both found the system to be transparent enough for easy use. Both expressed interest in using it again despite recording problems, though they both expressed interest in more consistency. Neither professor was bothered by having to wear a wireless microphone.

5. CONCLUSIONS

From our experience to date, PAOL is perceived by students to be a valuable tool that they would welcome in their classes. Without carrying out a more substantial deployment (more students, more classes), we cannot infer that it would have a strong impact on teaching and learning, but we have no reason to doubt its potential.

We achieved our goal to make capture transparent and compilation to a delivery system automatic. In the limited experiments to date, PAOL seems to capture the essence of the classroom experience and provide it through a relatively easy-to-use and acceptable interface. Because of its transparency, PAOL can be easily integrated into any course taught in our equipped lecture room with minimal effort by the lecturer. Our experience with MANIC and other content delivery systems demonstrates that instructors find new and innovative ways of using the technology once it is available. The cost of replicating this systems is within the range of typical classroom A/V equipment.

Our challenge is to increase the system’s consistency and reliability. Quality is meaningless if students cannot access the content easily on demand. Thus, to assess PAOL’s impact on teaching and learning, we need to consistently, reliably, and efficiently capture and compile the content into the delivery system so it is available for use by students and instructors. As we discovered, choosing an appropriate operating system may be as important as algorithm development to provide the consistency and reliability we require.

6. FUTURE WORK

We intend to record 2 courses in Spring 2009 and capture and distribute content to support a more substantial assessment of usability and an evaluation of the system’s impact on teaching and learning. One course will be a Web programming class for 60 non-computer science majors that uses a complete mix of projected, written, and demonstrated classroom materials that should test the limits of the system. It also will introduce PAOL to perhaps a more critical user community. The second course will be a typical computer science undergraduate class.

We are converting PAOL, the software, and sensors, to run under Linux, which we believe will provide a more robust and stable system capable of completely recording a class for an entire semester. We will continue to work on the GUI and capture algorithms based on student feedback throughout the semester to improve the system. We are putting in new video creation and compression to address the problems with streaming and downloading. We will have the opportunity to compare the Web programming and computer science courses with previous classes not supported by PAOL to begin to understand its impact on teaching and learning.

7. ACKNOWLEDGMENTS

This work was partially funded by the National Science Foundation under grants CNS-0749756 and CNS-0970833, and by the University of Massachusetts under an ITC Strategic Initiative Grant.

8. REFERENCES