ProjectorBox: Seamless presentation capture for classrooms

Laurent Denoue, David M. Hilbert, John Adcock, Daniel Billsus, and Matthew Cooper
FX Palo Alto Laboratory
Palo Alto, CA
United States
{denoue, hilbert, adcock, billsus, cooper }@fxpal.com

Abstract: Automatic lecture capture can help students, instructors, and educational institutions. Students can focus less on note-taking and more on what the instructor is saying. Instructors can provide access to lecture archives to help students study for exams and make-up missed classes. And online lecture recordings can be used to support distance learning. For these and other reasons, there has been great interest in automatically capturing classroom presentations. However, there is no simple solution that is completely automatic. ProjectorBox is our attempt to create a "zero user interaction" appliance that automatically captures, indexes, and manages presentation multimedia. It operates continuously to record the RGB information sent from presentation devices, such as an instructor's laptop, to display devices such as a projector. It seamlessly captures high-resolution slide images, text, and audio. A web-based user interface allows students to browse, search, replay, and export captured presentations.

Introduction

Instructors often hand out presentation slides or publish them online as study aids. However, this requires additional effort on the part of instructors and leads to materials that do not accurately reflect what occurred in the classroom. Inconsistency is a problem because instructors often change their slides after printing or uploading them. Inaccuracy is a problem because instructors often show slides in non-sequential order, focusing attention on some slides while skipping others entirely. And incompleteness is a problem because instructors' verbal explanations are absent from slides. For these and other reasons, there has been great interest in automatically capturing lectures.

Heavyweight solutions that leverage video cameras and room instrumentation (Abowd 1999, Mukhopadhyay and Smith 1999, Chiu et al. 2000) can produce rich lecture records, but are notoriously expensive to install, operate, and maintain. Lightweight software-based solutions capture media directly from the PC screen (Shinyama 2003, Ziewer 2004) or from specific presentation software packages (Denoue et al. 2004, Anystream 2005). However these solutions fail whenever non-preconfigured PCs, such as a guest lecturer's personal laptop, or unsupported presentation software is used. The constraints and limitations of existing systems led us to design and implement *ProjectorBox*, a turnkey solution for seamless lecture capture, indexing and reuse.

The goal of this work is to create a system that is lightweight and inexpensive, hardware and software independent (so that any presentation PC running any presentation software can take advantage of it), and that does not burden its users with unnecessary setup or maintenance overhead. The only solutions that can achieve this degree of platform independence and setup convenience are based on video signal interception. These systems intercept the RGB signal sent from PCs to projectors and create slide image records (SonicFoundry 2005) or screen video records (NCast 2005). However, despite their advantages, RGB-based solutions introduce considerable challenges. In order to generate high-quality presentation content suitable for indexing, retrieval and reuse, an RGB-based solution must infer high-level information from the video signal. For example, the system must be able to distinguish between slide and non-slide images, automatically detect boundaries between presentations, and extract text for indexing and retrieval. In this paper we describe how ProjectorBox addresses these challenges by pairing seamless video capture with intelligent media analysis to provide a solution that supports effortless lecture capture, indexing, and reuse.

Design Challenges

Our laboratory has pioneered a number of autonomous meeting capture capabilities (Chiu 2000). From this experience we observed great variability in the costs and benefits associated with different capture modalities, including issues in recording, retrieving, and reusing different media types. These previous experiments led us to focus on RGB capture coupled with media analysis and full-text indexing for improved retrieval and reuse. In this paper, we focus on four technical challenges that are inherent to the RGB-based design of ProjectorBox:

Detection of Presentation Content. Given our choice of an RGB-based approach, our first challenge was to automate capture so that presenters would not need to remember to start and stop recordings themselves. Researchers have already noted the importance of not distracting instructors with new recording technologies, particularly at the beginning and end of classes when students often approach them to ask questions (Abowd 1999). In terms of RGB capture, this meant we needed to robustly classify screen activity as either "associated with a presentation" or as desktop activity "not associated with a presentation". We briefly describe one of the slide classification algorithms we developed to address this challenge.

Presentation Segmentation. Because we envisioned a solution that would run continuously in a room used by multiple people for multiple presentations, we needed to automatically structure captured media into separate presentations.

Presentation Retrieval. Students want to retrieve lectures based on content and access captured media non-linearly, as opposed to having to play through sequential video (Abowd 1999). We experienced similar requirements in our own corporate environment (Chiu 2000). As a result, we apply optical character recognition (OCR) to slide images and create a full-text index to enable searching and non-linear access.

Presentation Export. Our system automatically creates a web-based archive that can be accessed by anyone authorized to use the system. However, our previous experience identified the need to export recordings so students can use them as study aids on their laptops while not connected to the Internet (Denoue et al. 2004). This led us to add the ability to export to a variety of file formats.

The following sections describe the ProjectorBox system, and how we address these challenges in more detail.

System description

System Components and Architecture

We implemented ProjectorBox as a PC-based system equipped with a high-resolution VGA capture card (DataPath 2005). This card can capture VGA signals from any computer at any resolution up to 1600x1200. In addition, ProjectorBox can capture audio streams using any Windows-compatible audio device. We installed our prototype system in a small-form-factor PC case, which is easy to deploy in classrooms and can be integrated with existing presentation podiums.

The ProjectorBox system consists of two main components: the *capture component* and the *server*. In addition to the video and audio capture hardware, the capture component consists of a software application that periodically transmits data to the server for further analysis and storage. Since instructors often flip back and forth through slides non-linearly, the capture component sends an image to the server only after it has been shown for several seconds. Based on real usage in the classroom, a 4 second interval produces reasonable results: slides that were not meant to be presented or discussed are omitted, while important presentation content is preserved. For each captured slide, a corresponding audio clip is recorded from an external microphone and stored as a compressed MP3 file. In our experience, a standard low-cost PC microphone can work as well as more expensive alternatives in a typical classroom.

While ProjectorBox is intended to run unobtrusively, it is important that presenters be able to control the recording process when necessary. For example, instructors may wish to disable slide and audio capture for "off the record" remarks or guest lectures with copyright restrictions. To this end, we installed a switch on the system's front panel to temporarily disable the capture component.

When the server receives an image, it generates a thumbnail version for the web interface and calls the OCR component to extract its textual content along with the bounding boxes for each word in the image. We currently use Microsoft's Document Imaging OCR because it can be easily automated by external applications. Finally, the data is time stamped and saved in a relational database. The server also performs slide classification and presentation segmentation, and provides a web-based user interface for retrieval, which we describe in the following sections.

The capture component transmits data to the server using HTTP to enable a flexible architecture in which the capture and server components can run on the same or separate PCs. For example, a single server can integrate content sent from lightweight capture components distributed in multiple classrooms and conference rooms.

One of our goals for ProjectorBox was to minimize bandwidth requirements. On data captured during one year, the average size of one hour of recording is 30MB (250 KB per minute for the MP3, and 400 KB per slide image, with 40 slides per hour). This is ten times lower than state of the art MPEG4 video encoders for similar high-resolution encodings (e.g. 1024x768 pixels).

Detection of Presentation Content

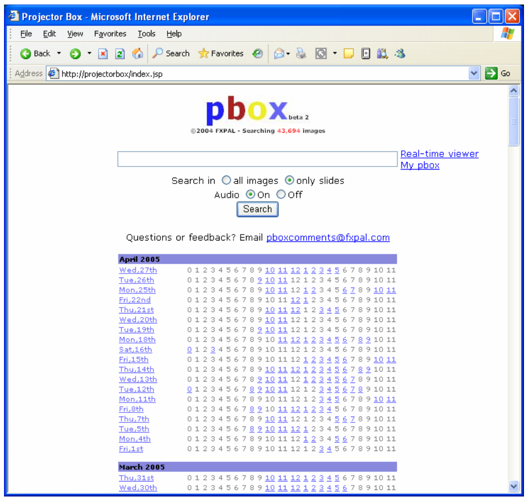
Since ProjectorBox runs 24/7, we have found that up to two thirds of the images captured are not slide images associated with a presentation. For example, a presentation is often preceded by images of the desktop and the presenter starting the presentation software and opening the presentation file. Clearly, these images should be omitted from a high-quality presentation archive. We address this problem using an automated image classification approach. The resulting classifier can reliably distinguish between "presentation" and "non presentation" content. As a first approximation to identifying presentation content, we focused on identifying slides. Slides are typically very different from normal desktop screens in that they mostly contain text in large fonts. In order to classify images as slide or non-slide images, we leverage the OCR data: we compute the average bounding box heights of the words and classify an image as a slide if the average is above a threshold. According to our experimental analysis, a threshold value of 2% of the height of the image gives the best results in terms of precision and recall.

Presentation Segmentation

In order to support retrieval of lectures as cohesive units that closely approximate the original presentation, we must be able to automatically group slides into presentations. We initially tried to leverage our slide classifier for presentation segmentation. In this initial approach, a presentation consisted of a series of images classified as slides, with the appearance of one or more non-slide images defining the presentation boundaries. However, this simple algorithm tended to over-segment whenever a slide was misclassified as a non-slide (e.g., if a slide contains a lot of text in a small font). Our new method relies on time: we create a new presentation if the time difference between two successive slide images is above a user-definable threshold. For our internal deployment, the default threshold is 20 minutes. The performance of this simple threshold emphasizes precision in segmentation over recall, that is, the results tend to under-segment the collection of slide images. Manual correction of the under-segmentation (presentation splitting) tends to be simpler and less tedious than correction of an over-segmentation (presentation merging). And under-segmented presentations (in which adjacent presentations are sometimes grouped together) seemed less annoying to users than over-segmented presentations (in which a single presentation was often broken into several parts). Nonetheless, we are actively experimenting with alternate approaches to this problem.

Presentation Retrieval

The previous capture and processing steps give us a series of presentations, consisting of individual slide images, corresponding audio clips and extracted text. We have implemented a web-based user interface in HTML and JavaScript that supports two methods for quickly retrieving content. The main page shows a list of dates and times, indicating when something has been captured (Figure 1, left). If the date and time of a specific lecture is known, this page provides a single-click solution to presentation retrieval.


Projector Box - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go
pbox beta 2
Real-time viewer
My pbox
Search

http://projectorbox/index.jsp

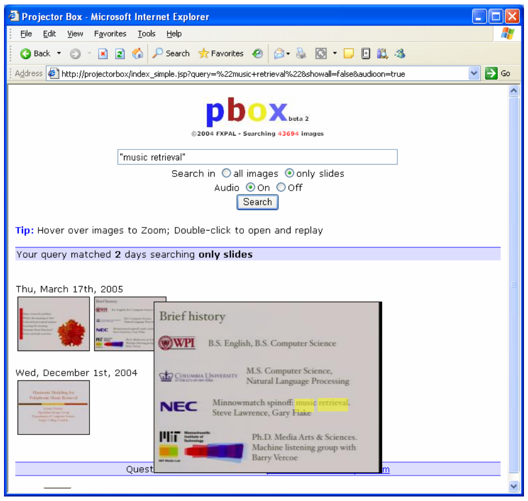
©2004 FXPAL - Searching 43,694 images

Search in ○ all images [✓ only slides]
Audio [✓ On] ○ Off

Questions or feedback? Email pboxcomments@fxpal.com

```
April 2005
Wed,27th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Tue,26th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Mon,25th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Fri,22nd     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Thu,21st     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Wed,20th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Tue,19th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Mon,18th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Sat,16th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Fri,15th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Thu,14th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Wed,13th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Tue,12th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Mon,11th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Fri,8th      0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Thu,7th      0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Tue,5th      0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Mon,4th      0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Fri,1st      0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11

March 2005
Thu,31st     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
Wed,30th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
```



http://projectorbox/index_simple.jsp?query=%22music+retrieval%22&showall=false&audioon=true

©2004 FXPAL - Searching 43694 images

"music retrieval"

Search in ○ all images [✓ only slides]
Audio [✓ On] ○ Off

Tip: Hover over images to Zoom; Double-click to open and replay

Your query matched **2** days searching **only slides**

Thu, March 17th, 2005

Wed, December 1st, 2004

Brief history

B.S. English, B.S. Computer Science

M.S. Computer Science, Natural Language Processing

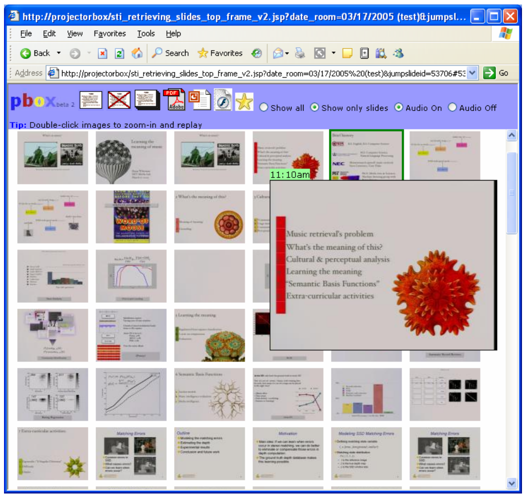
Minnowmatch spinoff: music retrieval, Steve Lawrence, Gary Flake

Ph.D. Media Arts & Sciences, Machine listening group with Barry Vercoe

Figure 1: The web interface allows students to select a specific date/time (left) or use full-text search (right).

In addition, the interface provides a text field for full-text search of all captured presentations, allowing students to retrieve classroom material as easily as retrieving a web page from Google. The result page shows matching slides organized by date, and query terms are automatically highlighted in yellow on the thumbnail to help the student understand why this image was retrieved (Figure 1, right). When the student identifies the relevant image, a single click brings up the day viewer (Figure 2).

The day viewer (Figure 2) shows thumbnail images of slides captured during a specific day. Moving the mouse over a slide shows an enlarged version of the slide and also plays the associated audio clip. This page provides a very easy way to locate a segment of interest that students might want to replay. Double-clicking on an image brings up a slide player that supports in-depth listening to the classroom material: students can quickly switch between all slides with previous and next buttons or arrow-keys and listen to corresponding audio clips.


http://projectorbox/sti_retrieving_slides_top_frame_v2.jsp?date_room=03/17/2005 (test)&jumpsl...
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go
pbox beta 2
PDF
Adobe
11:10am
WORD OF MOUSE

http://projectorbox/sti_retrieving_slides_top_frame_v2.jsp?date_room=03/17/2005%20(test)&jumpslideid=53706#53

○ Show all [✓ Show only slides] [✓ Audio On] ○ Audio Off

Tip: Double-click images to zoom-in and replay

Music retrieval's problem
What's the meaning of this?
Cultural & perceptual analysis
Learning the meaning
"Semantic Basis Functions"
Extra-curricular activities

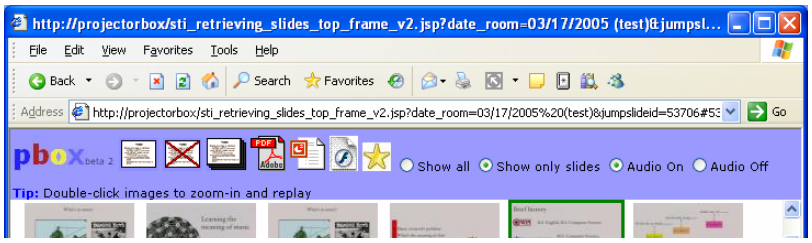
Figure 2: The day viewer shows all images captured during a specific day. Students can quickly browse the presentations by moving the mouse over specific images: an enlarged image is shown and the corresponding audio clip is played.

Because the algorithm for automatically classifying images as slide or non-slide is not perfect, the day viewer also allows students to quickly re-label images. Students can select an image or series of images and then click on the "is a slide" or "is not a slide" icon to quickly re-classify these images. Presentation boundaries can also be manually revised.

Exporting presentations

We learned from previous experiments with a note-taking system (Denoue et al. 2004) that hosting the content on a server is not enough: students need to be able to download the presentation to their personal desktops or laptops, so that lecture content can be accessed anytime and anywhere, independent of Internet availability.

Based on this feedback, we have implemented a very simple solution for exporting content to PDF, PowerPoint or Macromedia Flash. Students simply select images they would like to export (e.g. a whole presentation, individually selected slides or just a single slide) and then click on one of three export icons located on top of the day viewer page (Figure 3).



http://projectorbox/sti_retrieving_slides_top_frame_v2.jsp?date_room=03/17/2005 (test)&jumpsl...

http://projectorbox/sti_retrieving_slides_top_frame_v2.jsp?date_room=03/17/2005%20(test)&jumpslideid=53706#53

Show all [✓ Show only slides] [✓ Audio On] Audio Off

Tip: Double-click images to zoom-in and replay

Figure 3: The interface contains icons to export selected images as PDF, PowerPoint or Flash.

The generated PDF contains 2 images per page and is useful for printing the presentation. We support export to PowerPoint by automating PowerPoint on the server: the server generates a new slide for each image selected by the student. This format is useful for students who are used to taking notes in PowerPoint. They can also print the document in all different formatting options provided by PowerPoint. To support Flash export, we dynamically create an *swf* file using the open source *Ming* library (Ming 2005) on the server: the resulting animation contains slide images and corresponding audio clips, allowing students to view slides and listen to the instructor.

Finally, the web interface contains a button to bookmark specific images: this allows students to build a personal repository of particularly useful slides ("MyPBox") to simplify subsequent access to relevant presentation content.

Deployments and usage

We believe that deploying research systems in real-world settings is the best way to observe their impact on users and identify user needs (Trevor et al. 2002, Hilbert and Trevor 2004). Thus, as soon as we had a working prototype, we deployed it in our corporate conference room. The system has been in use for over a year and has captured over 40,000 images. This deployment has provided us with data for improving our slide classification and segmentation algorithms. After making several improvements, we deployed ProjectorBox in two universities. In addition to collecting image data to improve our algorithms, we are collecting web access logs, and plan to survey and interview students. Our goal is to collect data and feedback to improve ProjectorBox's support for both corporate and educational needs. In this paper we report only preliminary findings.

At the Naval Postgraduate School (NPS) in Monterey, we are working with instructors who are interested in automatic slide capture and note-taking technologies. They adopted ProjectorBox enthusiastically and have already recorded lectures from one of their courses. Installing ProjectorBox in their classroom was straightforward. However, the instructors decided they would rather not leave it in the classroom because this would require them to negotiate with the support personnel responsible for maintaining the classroom PCs. Thus, they simply carry the unit into the classroom for capture, and carry it back to their office after class where they connect it to the Internet to serve archive requests. While we believe an appliance-based solution is well-suited for many corporate and educational settings, a software solution installed on the instructor's laptop may have been more convenient in this particular case. Furthermore, a software capture solution would nicely complement an appliance-based solution in corporate settings where knowledge workers not only want to record in-house presentations, but also presentations they themselves give while on the road. Thus, we are considering adding a software-based mobile capture component that can be installed on laptops that will subsequently transmit captured data to a central archive server.

At San Francisco State University we are working with educational technologists who created a computer-augmented classroom for experimenting with new educational technologies (SFSU 2005). In this case, our adopters are technologists in charge of classroom technology, as opposed to the instructors themselves. This led to an unexpected result: namely, some instructors turn ProjectorBox off just before giving their lecture. We believe this is because they were only informed of ProjectorBox's existence (and its simple console for suspending recording) but were not informed of the purpose of ProjectorBox and the benefits for them and their students. We will address this by meeting with the instructors and students themselves to explain ProjectorBox and our evaluation goals.

Conclusions and future work

In this paper we described the development and deployment of ProjectorBox, a turnkey appliance for automatically capturing presentations. It operates continuously to record the RGB information sent from presentation devices, such as a presenter's laptop, to display devices, such as a projector. The RBG-based approach introduces several technical challenges including: detection of presentation content, presentation segmentation, presentation retrieval and presentation export. We explained how we addressed these challenges based on data collected from our corporate conference room over a one year trial period.

We plan to collect and analyze more data from our university deployments to help guide future research. We have already identified some desirable new features based on early feedback. Several presenters and users have requested that dynamic content—e.g., when the presenter shows videos, animations, web pages, and software demonstrations—be captured as video in addition to the static slide content we currently capture. This will require new algorithms for robustly detecting dynamic presentation content (as opposed to dynamic desktop interactions not associated with any presentation) and storage of video clips. We also plan to investigate support for allowing students to link their notes to captured slides and audio (both during and after class) as requested by our NPS collaborators.

References

1. Abowd, G.D. (1999). Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal 38,* pp. 508-530.
2. Anystream (2005). Apreso: http://www.apreso.com
3. Chiu, P., Kapuskar, A., Reitmeier, S., and Wilcox, L. (2000). Room with a Rear View: Meeting Capture in a Multimedia Conference Room. *IEEE MultiMedia Magazine, 7(4),* Oct-Dec 2000, pp. 48-54.
4. DataPath (2005). Vision RBG-PRO: http://www.datapath.co.uk/visRGBPRO.htm
5. Denoue L., Singh G., and Das A. (2004). Taking Notes on PDAs with Shared Text Input. *ED-MEDIA 2004,* Lugano, Switzerland.
6. Hilbert, D.M. and Trevor, J. (2004). Personalizing Shared Ubiquitous Devices. *Interactions Magazine, 11(3).*
7. Ming (2004). A SWF output library and PHP module: http://ming.sourceforge.net
8. Mukhopadhyay S. and Smith, B. (1999). Passive capture and structuring of lectures. *Proceedings of the seventh ACM international conference on Multimedia,* Orlando, Florida, pp. 477-487.
9. Ncast (2005). TelePresenter: http://www.ncast.com/telepresenter.html
10. SFSU (2005). Center for the Enhancement of Teaching: http://www.cet.sfsu.edu/
11. Shinyama, Y. (2003). vnc2swf: http://www.unixuser.org/~euske/vnc2swf/
12. Sonic Foundry (2005). MediaSite: http://www.mediasite.com
13. Trevor, J., Hilbert D.M., and Schilit, B.N. (2002). Issues in Personalizing Shared Ubiquitous Devices. *Proceedings of the Fourth International Conference on Ubiquitous Computing*, Göteborg, Sweden.
14. Ziewer, P (2004). Navigational Indices and Full-Text Search by Automated Analyses of Screen Recorded Data. *E-Learn 2004,* Washington, D.C., pp. 3055-3062.