Seamless presentation capture, indexing, and management

David M. Hilbert, Matthew Cooper, Laurent Denoue, John Adcock, and Daniel Billsus
FX Palo Alto Laboratory, Inc., 3400 Hillview Ave., Bldg. 4, Palo Alto, CA, USA 94304
{hilbert, cooper, denoue, adcock, billsus}@fxpal.com

ABSTRACT

Technology abounds for capturing presentations. However, no simple solution exists that is completely automatic. ProjectorBox is 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 a presenter's laptop, to display devices, such as a projector. It seamlessly captures high-resolution slide images, text and audio. It requires no operator, specialized software, or changes to current presentation practice. Automatic media analysis is used to detect presentation content and segment presentations. The analysis substantially enhances the web-based user interface for browsing, searching, and exporting captured presentations. ProjectorBox has been in use for over a year in our corporate conference room, and has been deployed in two universities. Our goal is to develop automatic capture services that address both corporate and educational needs.

Keywords: Presentation capture, indexing, retrieval

INTRODUCTION

Multimedia presentations are ubiquitous in education, business, and government. But presentation archives are rare due to installation, operation, and maintenance costs. As a result, useful information passes through projectors all the time and is lost. If we could create useful archives automatically—*without any added burden on anyone*—the benefits would be far reaching. In education it would mean support for distance learning, digital study aids, and a way to make-up missed classes. In business it would mean support for corporate communications, corporate memory, and a means for viewing missed meetings. And in conference settings it would mean support for automatic digital proceedings. For these and other reasons, automatic capture of presentations and meetings has received significant attention in research and industry. However, no simple solution exists that is completely automatic.

Heavyweight solutions that leverage video cameras and room instrumentation[1,3,8] can produce rich presentation records, but are notoriously expensive to install, operate, and maintain. Lightweight software-based solutions capture media directly from the PC screen[11,13] or from specific presentation software packages[2,5]. However these solutions fail whenever non-preconfigured PCs, such as a presenter’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 presentation capture, indexing, and management.

Our objective 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 minimally burdensome in terms of setup and maintenance overhead. The only solutions that can achieve this degree of platform independence and convenience are based on video signal interception. These systems intercept the RGB signal sent from PCs to projectors and create slide image records[12] or screen video records[9]. 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 support presentation capture, indexing, and retrieval.

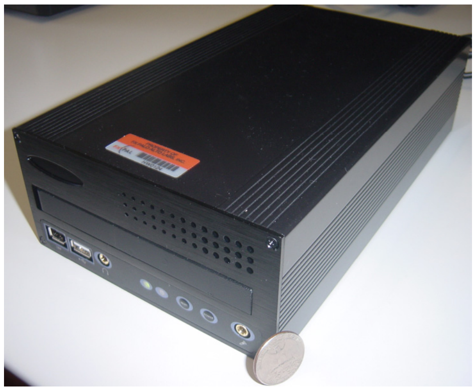
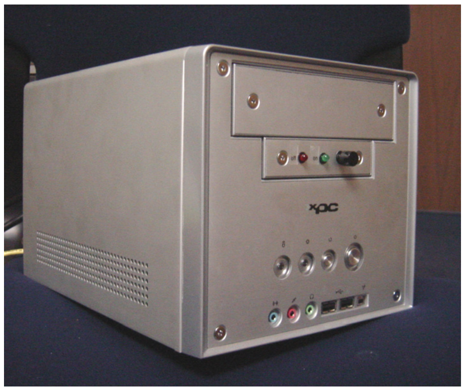



Figure 1: Two ProjectorBox hardware prototypes. The prototype on the right includes a custom front panel indicator and switch to temporarily disable recording for "off the record" comments and lectures with copyright restrictions.

1.1. Design Challenges

Our laboratory has pioneered a number of autonomous meeting capture capabilities[3]. From this experience, we have observed significant tradeoffs between the costs and benefits associated with different capture modalities, specifically with recording, retrieving, and reusing different media types. These previous experiments led us to focus on RGB capture coupled with image analysis and full-text indexing for improved retrieval and reuse. In this paper, we focus on three technical challenges that are inherent to the RGB-based design of ProjectorBox:

Detection of Presentation Content. The first challenge is to automate capture so that presenters do not need to start and stop recordings themselves. Researchers have already noted the importance of not distracting lecturers with new recording technologies, particularly at the beginning and end of lectures when audience members ask questions[1]. In terms of RGB capture, this requires robust classification of screen activity as either "associated with a presentation" or as desktop activity "not associated with a presentation". We present several slide classification algorithms designed to address this challenge.

Presentation Segmentation. We envision a solution that runs continuously in a room used by multiple people for multiple presentations. Thus, we need 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[1]. We experienced similar requirements in our own corporate environment[3]. 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.

1.2. Paper Overview

In Section 2, we describe the components and architecture of the ProjectorBox system. Section 3 details experiments in classification of slide and non-slide content captured from presentations. We have experimented with combinations of different classification algorithms and several different low-level features for automatic detection of presentation content. Section 4 describes a simple, pragmatic approach to presentation segmentation. Section 5 describes the user interface for presentation retrieval. We conclude by discussing our experiences deploying the system in multiple settings, and future directions for system improvements.

2. SYSTEM DESCRIPTION

We implemented ProjectorBox as a PC-based system equipped with a high-resolution VGA capture card[4]. 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 have installed our prototype in two different small-form-factor PC cases, which are easy to deploy in classrooms and conference rooms and can be integrated with existing presentation podiums (Figure 1).

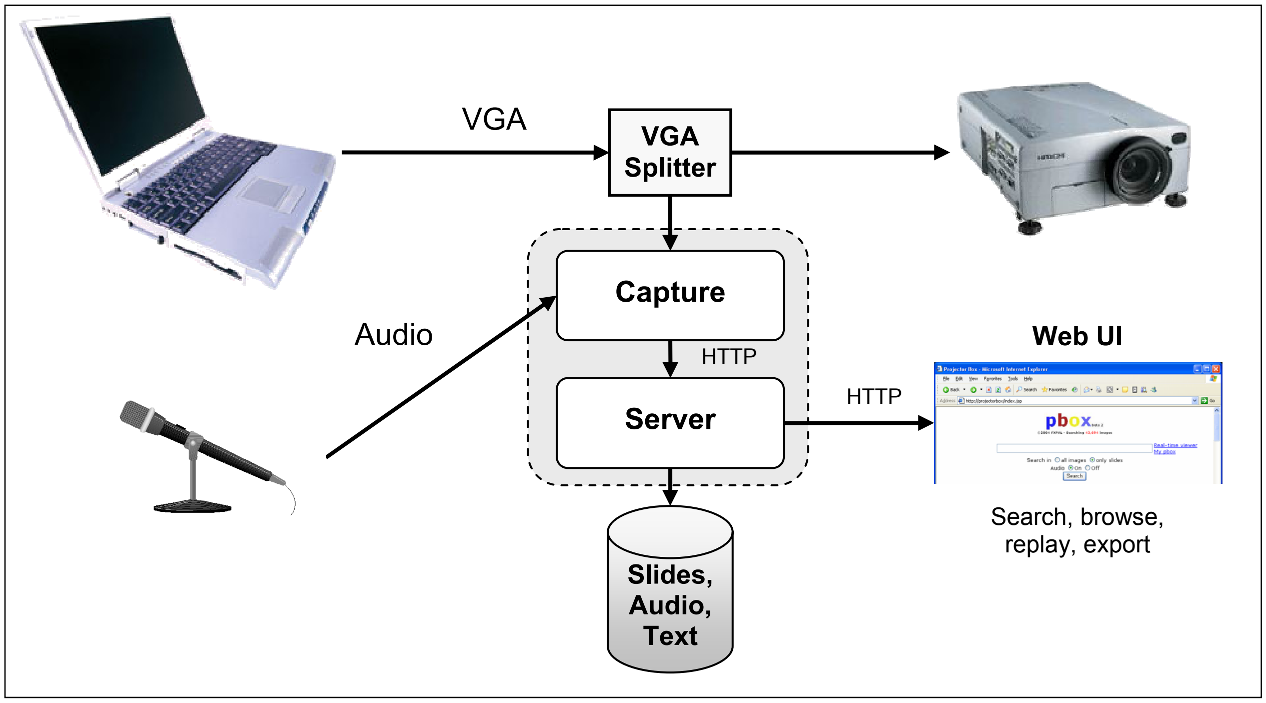


Figure 2: Illustration of the ProjectorBox system.

The ProjectorBox system consists of two main components: the *capture component* and the *server* (Figure 2). 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 presenters 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 usage in our corporate conference room, 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 image, a corresponding audio clip is recorded from an external microphone and stored as a compressed MP3 file. In our conference room, we use a ceiling microphone positioned above the center of the room. In our other deployments, we have used centrally positioned table microphones and standard low-cost PC microphones.

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).

3. DETECTION OF PRESENTATION CONTENT

In our corporate conference room, ProjectorBox runs 24/7. We have found that up to two thirds of the images captured are not slide images. 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 hidden from users perusing presentation content. 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.

The main feature used in the classification is the height of the text bounding boxes detected by the optical character recognition (OCR) process. The intuition is that most presentation slides use larger fonts than typically appear on a computer desktop or in other applications. As a result, bounding box heights are a useful cue for detecting slide images. Figure 3 shows mean histograms of text bounding box heights for slide and non-slide classes. The heights increase with the histogram bin index along the X-axis. The histogram for the non-slide class is sharply peaked for smaller bounding-box heights, while the slide class exhibits a more uniform histogram.

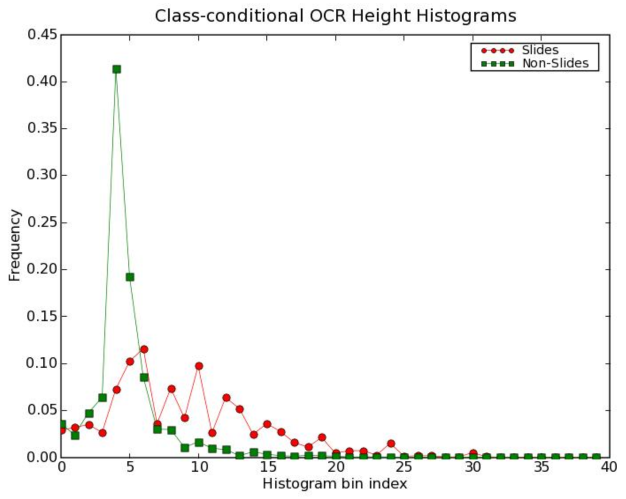


Figure 3: Mean histograms of bounding box height for slide images (red circles) and non-slide images (green squares).

One simple and effective approach to this problem is to compute the mean bounding box height over the entire OCR output, and compare it to a threshold. To compute the appropriate threshold, we process training data. Given training data indexed from *1,…,* N_{train}, denote the average bounding box height of image *i* by h_i. The threshold we have used is the global per-image mean:

$$\tau = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} h_i \quad . \tag{1}$$

For our training data, $\tau = 19.52$ pixels. The classification procedure simply compares the mean height of the bounding boxes extracted from a test image and compares it to the threshold, τ. If the mean height of a test image, h_{test} is greater than τ, the image is labeled as a slide. Otherwise, the image is labeled as a non-slide.

3.1. Slide image classification with bounding box height histograms

A common non-parametric classifier is the k-nearest-neighbor (kNN) classifier[6]. Given a set of labeled training data, and a test image, the first step is to determine the *k* elements of the training set that are closest to the test image in the feature space. Then, assign the majority class label among those *k* training images to the test image. We used histograms of the bounding box heights as features, and experimented with several distance measures for determining the nearest neighbors. Denote the histogram of the i^{th} slide by the vector X_i, with bin-indexed values $X_i(b)$. We show results using the chi-square similarity measure:

$$d_{chi}(X_i, X_j) = \sum_b \frac{(X_i(b) - X_j(b))^2}{X_i(b) + X_j(b)}, \tag{2}$$

the L-1 distance:

$$d_{L1}(X_i, X_j) = \sum_b |X_i(b) - X_j(b)|, \tag{3}$$

and the L-2 distance:

$$d_{L2}(X_i, X_j) = \sqrt{\sum_b (X_i(b) - X_j(b))^2}. \tag{4}$$

The tradeoff between precision and recall can be adjusted according to the application context. For testing, we control the sensitivity of the kNN classification using an integer parameter κ: $1 \le \kappa \le k$. For a given test vector, if at least κ out of the *k* nearest neighbors in the training data are from the slide class, we label the image as a slide and otherwise label it as a non-slide. κ is varied to produce full recall-precision curves.

3.2. Slide image classification with color features

We also tested kNN classification of images as slides/non-slides using image histograms. For this, we extract four luminance histograms from the border regions of each image. In each case we use the 10 rows/columns along the top, bottom, left, and right and compute a histogram of the luminance values. The resulting average histograms for both classes for all four regions appear in Figure 4. For all four regions, the luminance histogram is peaked at the higher end of the range for the slide class, relative to the non-slide class.

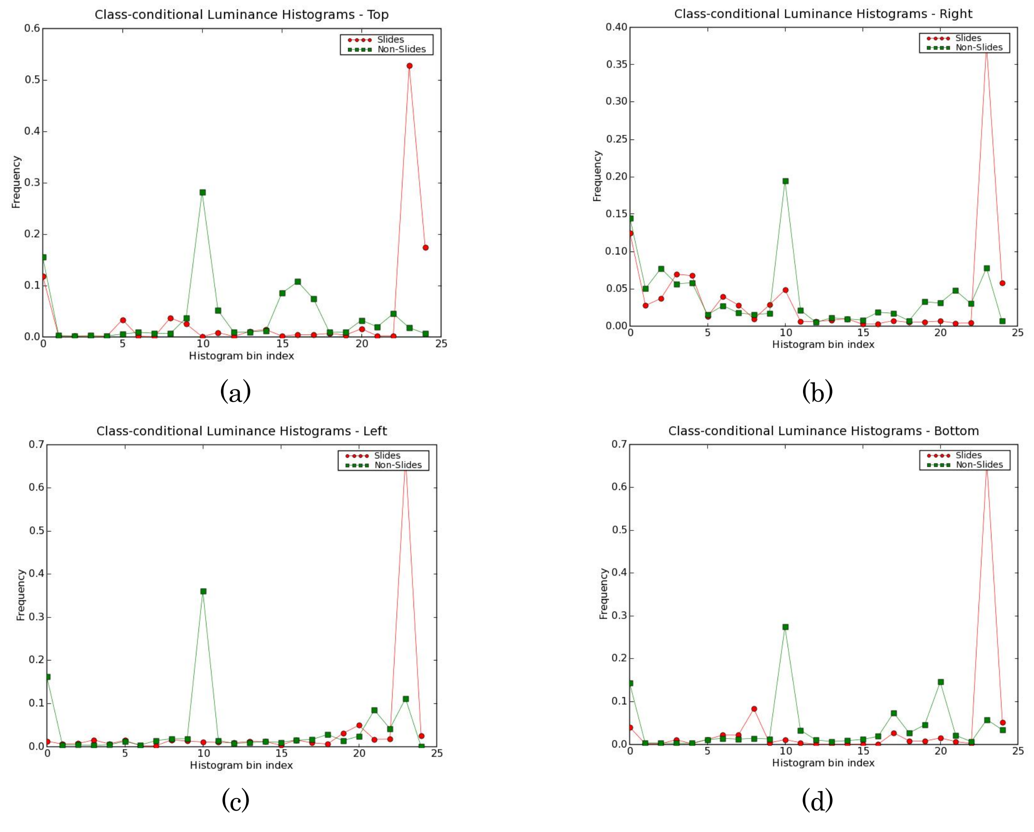


Figure 4: This figure shows average luminance histograms for slide images (red circles) and non-slide images (green squares) over the entire test database. Each panel corresponds to the 10 rows/columns of pixels along the top (a), right (b), left (c), and bottom (d).

3.3. Performance evaluation

For testing we compare several supervised methods using over five months of captured imagery comprising 20,849 images. 3553 image were labeled as slides. We divided the data into training and test sets. The first 5212 (25%) of the data is used for training, and the remaining 15,636 images are used for testing. To assess performance we compute two common figures of merit. Precision is a measure that penalizes false positives: non-slide images that are labeled as slides. Recall is a measure that penalizes false negatives: slide images that are labeled as non-slides. They are defined as:

$$\text{Precison} = \frac{\#\text{Correct detections of slides}}{\#\text{detections of slides}},$$
$$\text{Recall} = \frac{\#\text{Correct detections of slides}}{\#\text{of true slides}}.$$

For the kNN methods, $k = 11$.

The results of the thresholding approach appear in Figure 5 as the solid curve with points marked "X". For τ computed as in (1), the resulting precision is 0.743558 and recall is 0.844638. The curves for the kNN classifier with OCR features appear with marks "▲" (L-2), "●" (L-1), and "+" (chi-squared). Using each of the three similarity measures, classification performance is worse than using the simple threshold at high levels of recall. At high precision and lower recall, the kNN methods outperform the simple threshold. The upper curve ("■") shows the performance of the kNN classifier using the luminance histogram features and the chi-squared similarity measure. This method outperforms the OCR-based techniques.

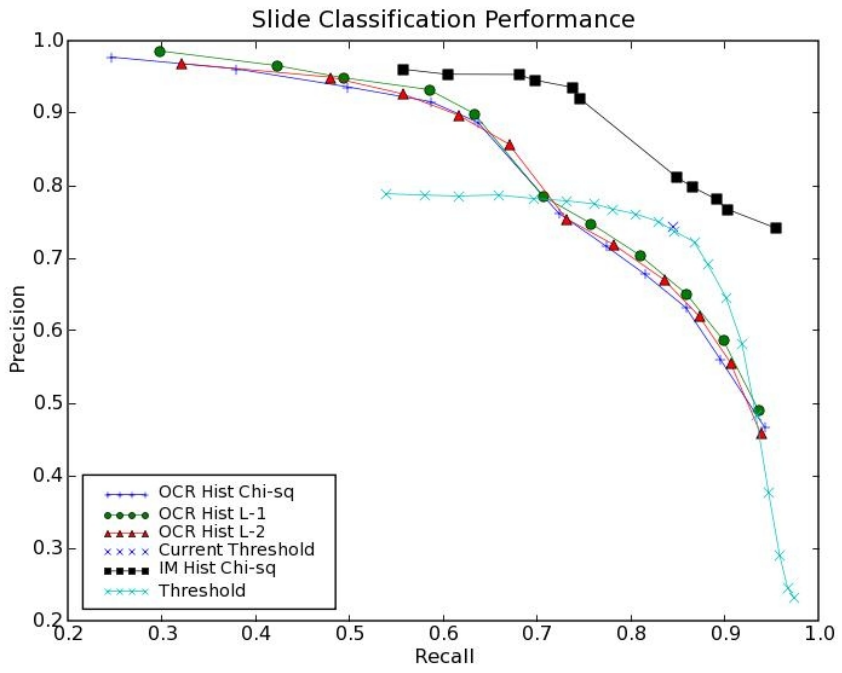


Figure 5: This figure shows performance curves for the various slide image classification techniques.

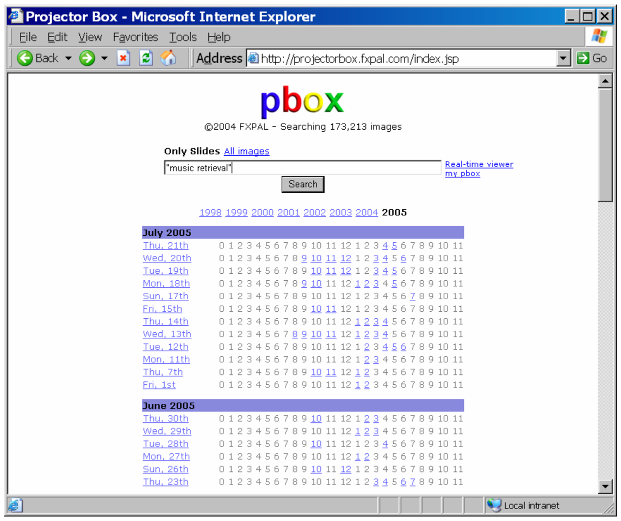
4. PRESENTATION SEGMENTATION

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 consists 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 tends to over-segment (note that the threshold-based slide classification is lower in precision). Our new method uses post-processing based on time: we delete presentation boundaries between two successive images if the time difference between them is below 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.

5. PRESENTATION RETRIEVAL

The previous capture and processing steps give us a series of presentations, consisting of individual slide images with 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 content has been captured (Figure 6 left). If the date and time of a specific presentation is known, this page provides a single-click solution to presentation retrieval.

In addition, the interface provides a text field for full-text search of all captured presentations, allowing users to easily retrieve materials. The result page shows matching slides organized by date, and query terms are automatically highlighted in yellow on the thumbnail to indicate why this image was retrieved (Figure 6 right). When the user identifies the relevant image, a single click brings up the day viewer (Figure 7 left).


Projector Box - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://projectorbox.fxpal.com/index.jsp
Go
pbox
Only Slides All images
Search
Real-time viewer
my pbox
Local intranet

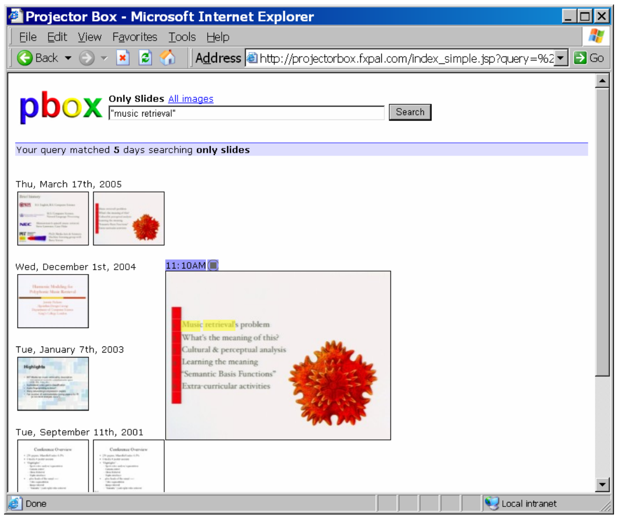
©2004 FXPAL - Searching 173,213 images

"music retrieval"

1998 1999 2000 2001 2002 2003 2004 **2005**

```
July 2005
Thu, 21th     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
Sun, 17th     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
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
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

June 2005
Thu, 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
Wed, 29th     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, 28th     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, 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
Sun, 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
Thu, 23th     0 1 2 3 4 5 6 7 8 9 10 11 12 1 2 3 4 5 6 7 8 9 10 11
```


Projector Box - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://projectorbox.fxpal.com/index_simple.jsp?query=%2
Go
pbox
Only Slides All images
Search
11:10AM
Done
Local intranet

"music retrieval"

Your query matched **5** days searching **only slides**

Thu, March 17th, 2005

Wed, December 1st, 2004

Tue, January 7th, 2003

Tue, September 11th, 2001

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

Figure 6: The web interface allows users to select a specific date/time (left) or use full-text search (right).

The day viewer (Figure 7 left) 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 users might want to replay. Double-clicking on an image brings up a slide player (Figure 7 right) that supports in-depth listening to the presentation material: users can quickly switch between all slides with previous and next buttons or arrow-keys and listen to corresponding audio clips.

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

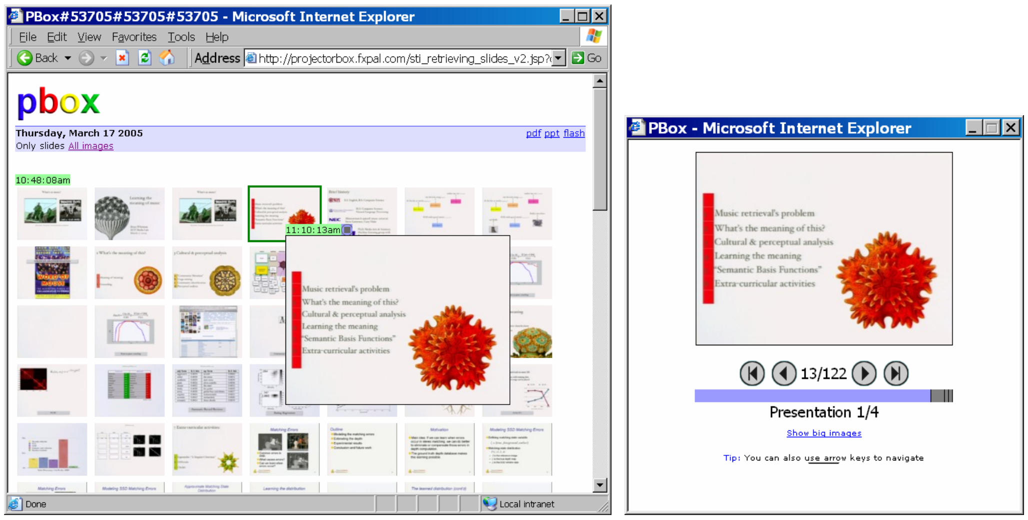

PBox#53705#53705#53705 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://projectorbox.fxpal.com/sti_retrieving_slides_v2.jsp?
Go
pbox
Thursday, March 17 2005
pdf ppt flash
Only slides All images
10:48:08am
11:10:13am
Music retrieval's problem
What's the meaning of this?
Cultural & perceptual analysis
Learning the meaning
"Semantic Basis Functions"
Extra-curricular activities
Done
Local intranet
PBox - Microsoft Internet Explorer
13/122
Presentation 1/4
Show big images
Tip: You can also use arrow keys to navigate

Figure 7: The day viewer (left) shows all images captured during a specific day. Users can quickly browse presentations by moving the mouse over specific images: an enlarged image is shown and the corresponding audio clip is played. Double-clicking on an image brings up the slide player (right) for playing back presentations sequentially or skipping backward and forward through slides.

6. 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[7]. 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. The primary internal use of the system has been to review presentation materials from meetings users were unable to attend. Users have also found the system convenient for recalling details from presentations given by important external visitors. Additionally, the system has been used to provide "talking points" for follow-up discussions after meetings, when it's helpful to directly reference presentation content.

After making several improvements, we deployed ProjectorBox in two university classrooms, at San Francisco State University (SFSU) and the Naval Postgraduate School (NPS) in Monterey. At SFSU we are working with educational technologists who created a computer-augmented classroom for experimenting with new educational technologies[10]. At NPS we are working directly with instructors who are interested in both automatic slide capture and note-taking technologies. They adopted ProjectorBox enthusiastically and have already recorded lectures from one of their courses. In addition to collecting image data to improve our algorithms, we are collecting web access logs, and are surveying and interviewing instructors and students to gain more insight. Initial results indicate students used ProjectorBox to review missed classes and study for exams, and would like to use ProjectorBox in future courses. Our goal is to collect more data and feedback to improve ProjectorBox's support for both corporate and educational needs.

7. CONCLUSIONS

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 RGB-based approach introduces several technical challenges including: detection of presentation content, presentation segmentation, and presentation retrieval. We explained how we addressed these challenges based on data collected from multiple real-world deployments.

Our automatic analysis components remain the subject of continuing research. We have currently deployed the threshold-based slide image classification algorithm, which provides the best tradeoff between performance and computational complexity. We initially used the results of this slide classification step to segment groups of contiguous slides into presentations. However, after anecdotally observing that users find merging over-segmented presentations more aggravating than splitting under-segmented presentations, we opted for a segmentation approach that favors precision over recall. Since our current algorithms are fast, we plan to experiment with exposing the threshold in the user interface (e.g., as a slider) to allow users to quickly vary the sensitivity of the slide classification and granularity of the presentation segmentation according to their preferences. We are presently experimenting with training models off-line for slide and non-slide imagery to leverage the considerable available labeled data at training time, rather than at classification time (as with the kNN classifier). Similarly, we are studying the characteristics of groups of slides within presentations to construct models for presentation segments and non-presentation segments. Finally, we are also experimenting with methods for joint slide classification and presentation segmentation.

We plan to collect and analyze more data from our 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).

REFERENCES

1. Abowd, G.D., Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal 38,* pp. 508-530, 1999.
2. Anystream. Apreso: http://www.apreso.com , 2005.
3. Chiu, P., Kapuskar, A., Reitmeier, S., and Wilcox, L., 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. Vision RBG-PRO: http://www.datapath.co.uk/visRGBPRO.htm , 2005.
5. Denoue L., Singh G., and Das A., Taking Notes on PDAs with Shared Text Input. ED-MEDIA 2004, Lugano, Switzerland, 2004.
6. Duda, R. and Hart, P., *Pattern Classification and Scene Analysis*. Wiley & Sons, New York, 1975.
7. Hilbert, D.M. and Trevor, J. Personalizing Shared Ubiquitous Devices. *Interactions Magazine, 11(3)*, 2004.
8. Mukhopadhyay S. and Smith, B., Passive capture and structuring of lectures. Proceedings of the seventh ACM international conference on Multimedia, Orlando, Florida, pp. 477-487, 1999.
9. Ncast. TelePresenter: http://www.ncast.com/telepresenter.html , 2005.
10. SFSU. Center for the Enhancement of Teaching: http://www.cet.sfsu.edu/ , 2005.
11. Shinyama, Y., vnc2swf: http://www.unixuser.org/~euske/vnc2swf/ , 2003.
12. Sonic Foundry. MediaSite: http://www.mediasite.com , 2005.
13. Ziewer, P., Navigational Indices and Full-Text Search by Automated Analyses of Screen Recorded Data. E-Learn 2004, Washington, D.C., pp. 3055-3062, 2004.