

# Linking Multimedia Presentations with their Symbolic Source Documents: Algorithm and Applications

Berna Erol, Jonathan J. Hull and Dar-Shyang Lee

RICOH Innovations Inc., California Research Center  
2882 Sand Hill Road, Menlo Park, CA 94025  
+1-650-496-5700  
{berna,hull,dsl}@rii.ricoh.com

## ABSTRACT

An algorithm is presented that automatically matches images of presentation slides to the symbolic source file (e.g., PowerPoint™ or Acrobat™) from which they were generated. The images are captured either by tapping the video output from a laptop connected to a projector or by taking a picture of what's displayed on the screen in a conference room. The matching algorithm extracts features from the image data, including OCR output, edges, projection profiles, and layout and determines the symbolic file that contains the most similar collection of features. This algorithm enables several unique applications for enhancing a meeting in real-time and accessing the audio and video that were recorded while a presentation was being given. These applications include the simultaneous translation of presentation slides during a meeting, linking video clips inside a PowerPoint file that show how each slide was described by the presenter, and retrieving presentation recordings using digital camera images as queries.

## Categories and Subject Descriptors

H.3.4 [Information Systems]: Information Storage and Retrieval – systems and software.

## General Terms

Algorithms, Performance, Design, Experimentation.

## Keywords

E-learning, presentation recording, meeting recording, multimedia meeting room, document linking, synchronization

## 1. INTRODUCTION

Recorded presentations have great value for both reminding and communicating. In an educational setting, students are often enthusiastic users of recorded presentations. If they attend a lecture, they refresh their memory about specific points raised in

class by consulting an audio-visual record. They sometimes rely on the fact that a presentation is being recorded and don't attend a class in person. Instead, they either watch the presentation online in real-time or replay the recording later, perhaps the night before an exam. In the corporate world, multimedia presentations are still used for education and communication, but usage conditions and needs of the presenters and the audience may be different than those of students. For example, the audience in a presentation may not be fluent in the presenter's language and communicating some key points may be crucial for the business. Office workers frequently share presentation material and often need to give presentations for someone else. Also, many office workers prepare reports about the presentations they attended. Even though many tools were developed for classroom use of presentation recordings [1]-[3], limited work has been done on providing tools for effective utilization of captured presentations in a corporate setting.

Systems for recording presentations are becoming commonly available. Commercial solutions include authoring tools that let users create online presentations by recording audio, video, and presentation slides while a talk is being given [4][5]. Various research prototypes have been proposed that synchronize video of the speaker with the images of the slides they present [6][7]. Typically, a replay interface is provided that allows a viewer to click on the image of a presentation slide and be automatically forwarded to the corresponding point in the video. These systems enable retrieval and replay of recorded presentations.

We also developed a meeting and presentation capture system [8]. In our system, presenter's slides are captured and synchronized automatically with the captured audiovisual stream and whiteboard images, without requiring the presenter to install any software on their computer. After a presentation, a SMIL file is generated automatically and users can access the recording with a replay interface that provides random access via thumbnails of the presented slide images. A web-based search interface allows for retrieval of recorded presentations by time, location, and keywords.

After using our presentation system on a regular basis for almost two years, we observed that providing a powerful web-based search and access interface is crucial, but not sufficient for the best utilization of captured presentations by most office workers. This should be expected since like a student who needs to pass an exam, most office workers would only utilize these recordings if they make their work more efficient. We determined that by

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM Multimedia '03, Nov. 2-8, 2003, Berkeley, CA.

Copyright 2003 ACM 1-58113-722-2/03/0011 ...\$5.00.

linking captured slides and audiovisual recordings to the symbolic presentation file (e.g., PowerPoint file), we could provide various applications that could improve an office worker's efficiency and the usefulness of multimedia recordings of meetings and presentations.

One example application we implemented is *automatic simultaneous translation* of presentation slides. In this application, the presenter's slides are displayed in real-time in several different languages. This is achieved by automatically linking the presenter's projected slide to the symbolic PowerPoint file that it comes from and using the symbolic file as an input to a PowerPoint translation package. The translated slides are displayed in real-time either on another projector or on a web site that can be viewed on a participant's laptop. Considering that in the corporate world the audience of a presentation can contain people from different countries; this application can be very useful in communicating with people who are competent in different languages.

Another new application we implemented that is enabled by our linking technique is the embedding of video clips that show the presenter describing each slide directly in the symbolic source file. Office workers commonly share presentation slides with each other, modify and create new presentations using existing ones. Most of the time, presentation slides have little text on them and they require explanation from the person who prepared the slides. By linking audiovisual recordings to the presentation's symbolic source, it is possible to embed these recordings in the editable source file. This lets the receiver modify the presentation slides and see exactly how the original author described the material, thus significantly improving the stand-alone ability of the source file to communicate the presenter's message.

This paper describes a novel algorithm that enables the unique capabilities described above by matching images of presentation slides to the pages in a symbolic source file, which is in the case a PowerPoint file. Presentation slides are generally a combination of natural, synthetic images, and text. Our method is based on a combination of edge histogram analysis, string matching, layout analysis, and line profile matching techniques. Our experiments show that this method outperforms other techniques available in the literature in retrieval accuracy. Also, it is more flexible as it does not put any restriction on the kind of slide images it can process, and it is more robust to occlusions as it can handle partial matching.

Besides linking slide images to their symbolic source, we utilized the proposed linking technique for synchronizing slide images captured by various different devices, thus enabling other unique applications. For example, consider a case when a person takes a picture in a presentation. If the picture he takes contains a slide image, our slide linking technique can be used to associate that picture with the particular presentation and the moment when he took the picture. He can then playback the relevant presentation recording to refresh his memory or share the information with others.

In the next section we present some background work on presentation and meeting document linking applications and content-based linking techniques.

## 2. BACKGROUND

Most of the current presentation and lecture capture systems consists of a pan/tilt/zoom camera that is controlled either automatically or by an operator. There is usually one audiovisual stream generated and, at a given time, only the presentation slides, the speaker, or the audience is captured in a video sequence. In more sophisticated presentation recording systems, there are more than one camera capturing the same scene from different angles and presentation slides and whiteboard data can be captured in separated data streams. Synchronization and linking of these streams are critical for the efficient playback, retrieval, and access of the presentation recordings. This synchronization is typically done by capturing events such as the keystrokes at the presenter's computer or by time-based synchronization. The temporal coherency across multiple captured streams has been exploited by many researchers to facilitate access, visualization and analysis of the recordings. Time stamped events such as handwriting [9], notes [10], or browser activities [1][3] have proven to be very useful in providing easy access and effective indexing to audiovisual streams.

In some cases, synchronization/linking of presentation streams using only event capture or time-stamps is insufficient and content-based linking of meeting/presentation/lecture documents is required. In [11], Franklin et al., suggest linking presentation slides to the audio of the speaker by matching the speech content to the text content in the presentation slides. Another content-based linking method that is exploited by several researchers is the linking of visual streams through matching of the slide content captured in these streams. Mukhopadhyay et al. proposed in [2] to match the content of HTML pages that contain presentation slides to the low-resolution video that also includes the presentation slides. Their method is based on first dilating and binarizing the segmented slide images and frames to highlight the text regions, and then using the Hausdorff distance to compute the similarity between the text lines. Their method requires that the slide region be accurately segmented. Also, it works well only on slides that contain text. In [12], Chiu et al. proposed automatically linking multimedia data with a DCT-based image matching of the slide content. They propose to match the contents of scanned handouts, screen capture and presentation video. Their method is mostly suitable for matching high quality and high-resolution slide images and the performance of their method may degrade if the images are low-resolution and are not accurately segmented. Partial occlusion or the presence of blur also degrades its performance.

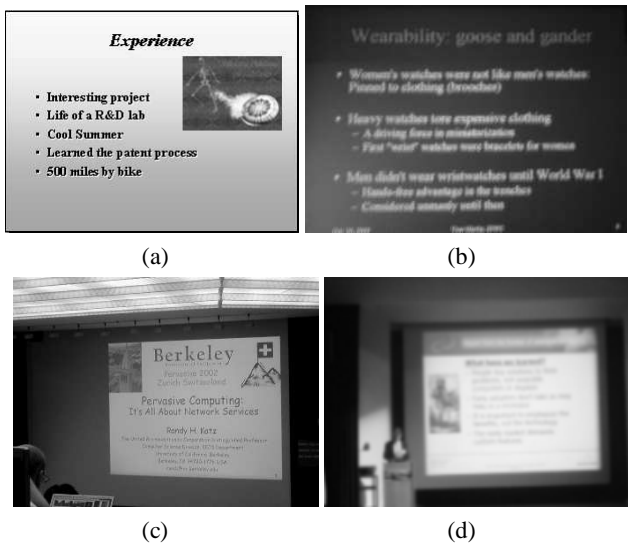
The rest of the paper is organized as follows. In the next section, we describe a new slide content matching algorithm, which overcomes the limitations of the algorithms described above. Section 4 gives an overview of the three new applications implemented based on our algorithm. In Section 5, the retrieval performance of our algorithm is presented and compared with the prior art. Finally, conclusions and future work are discussed in Section 6.

## 3. CONTENT-BASED LINKING

Slide images may be captured by a number of different devices, including video recorders, digital cameras, presentation capture

systems, scanners, document cameras, etc. Examples of slide images captured by various devices are presented in Figure 1. The image in Figure 1.a was obtained by saving the rendered presentation source file (in this case, PowerPoint) as a JPEG image, Figure 1.b and Figure 1.c are digital camera images, and Figure 1.d is a frame from a video recording.

We utilize a number of image features for slide content matching, based on the image capture device, target application, presence of blur, presence of occlusion, requirements for accuracy, etc. Slides from the same presentation typically have similar color histograms, dominant colors, etc. Therefore, most color features are not strong discriminators between these images. On the other hand, slides in a presentation contain different text, combinations of text lines, layouts, images, and graphics. Consequently, our experiments showed that the text content, edge content, as well as layout of a slide are strong discriminatory features among slide images that belong to the same presentation.



**Figure 1. Examples of different types of slide images. (a) Extracted from the original presentation file, (b) captured by a digital camera, includes only the slide region, (c) captured by a digital camera, includes the slide region as well as the surroundings, and (d) a frame captured by a video camera**

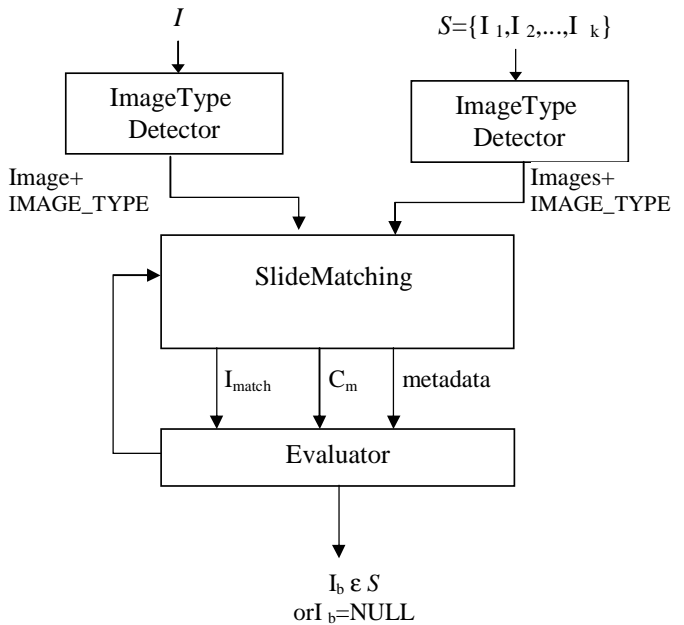
It is difficult, if not impossible, to find one image matching technique that can be used for every kind of slide image. Here, we employ a combination of edge histogram, string, layout, and text-line profile matching. Some of these methods are suitable for the analysis and retrieval of natural images, photos, graphics, etc., and others are useful for document image retrieval. Since slide images have the characteristics of document images and they can also contain graphics, photos, etc, our technique for slide image retrieval employs a combination of image matching methods.

The flow diagram of our generalized slide content matching technique is given in Figure 2. The algorithm takes an image,  $I$ , from a capture source and another set of images,  $S = \{I_1, I_2, \dots, I_k\}$ ,

from another source, and finds the best matching image in  $S$  to  $I$ . The functionality of each element in Figure 2 is explained below.

**Image type detector:** Slide images are classified into four groups to determine which content-based matching techniques are most suitable. This decision is based on the following two properties:

1. Image contains only the region of interest, i.e. a slide region  
Some images contain only a slide region, for example slide images obtained by capturing the presentation screen. Some other images may contain the slide region as well as the surroundings. For example, digital camera pictures of presentation slides, video frames of a presentation slide and presenter, etc.
2. Text-extractable image  
Most presentation slides contain text. In some cases, OCR can successfully extract the text in slide images. However, depending on the capture device, the capture resolution, font size/color in a presentation slide and the presence of blur, it may not be possible to obtain accurate text from every image.



**Figure 2. Flow of the generalized slide matching technique.**

In order to detect whether or not text can be extracted from a set of slide images, OCR is applied to several of the images in the set and the length of the extracted text and their confidence scores are reevaluated.

For classifying a slide image as containing only the Region of Interest (ROI), the image capture source can provide guidance. For example, a slide image obtained by screen capture is likely to contain only the ROI. On the other hand, a slide image captured by a digital camera may contain the ROI as well as the surroundings. Note that such images can always be segmented and skew, rotation, etc., can be corrected with post-processing.

obtain an accurate representation of the ROI. We do not address this problem in this paper.

Based on the above properties, slide images are classified as follows:

- Region of Interest (ROI) only with extractable text ( **ROI-Txt**), e.g. Figure 1.a
- ROI only without extractable text ( **ROI-N**), e.g. Figure 1.b
- Non segmented ROI with extractable text ( **N-Txt**), e.g. Figure 1.c
- Non segmented ROI without extractable text ( **N-N**), e.g. Figure 1.d

**Slide Matching** : Depending on the captured image type, capture devices, and the requirements of the target applications, slide content matching is performed using one or several combinations of the following content retrieval techniques.

- Edge histogram matching ( **EH**)
- Line profile matching ( **LP**)
- String matching ( **OCRS**)
- Layout matching ( **LY**)

Each matching technique outputs, for an input image  $I$ , the best matched image in a collection of slide images,  $S = \{I_1, I_2, \dots, I_k\}$ , similarity score,  $I_{match}$ , relative confidence score,  $C_m$ , and metadata. The individual matching techniques are explained in detail later in this section.

Table 1 shows the slide matching technique to be used for matching different kinds of slide images. The techniques listed in each cell can be used by themselves or in combination.

**Table 1. Slide matching technique to use based on image type.**

Image type	ROI-Txt	N-Txt	ROI-N	N-N
ROI-Txt	EH, LY, OCRS, LP	OCRS, LP	EH, LY, LP	LP
N-Txt	OCRS, LP	OCRS, LP	EH, LY, LP	LP
ROI-N	EH, LY, LP	EH, LY, LP	EH, LY, LP	LP
N-N	LP	LP	LP	LP

**$I_{match}$** : Best matched image to  $I$  in  $S = \{I_1, I_2, \dots, I_k\}$  based on the slide content matching technique.

**$C_m$ -Confidence measure**: Regardless of the matching method used, given an image,  $I$ , to be matched against a collection of slide images,  $S = \{I_1, I_2, \dots, I_k\}$ , a *relative match confidence score*,  $C_m$ , is computed based on the distance scores as follows:

$$C_m = \frac{(d_{\min 2} - d_{\min})}{d_{\min}},$$

where  $d_{\min}$  is the distance between  $I$  and the best matched image,  $I_b$ , in  $S$ ,  $d_{\min} = \min_{I_k \in S} \{d(I, I_k)\}$ , and  $d_{\min 2}$  is the

distance between  $I$  and the second best matched image in  $S$ ,  $d_{\min 2} = \min_{I_k \in \{S - I_b\}} \{d(I, I_k)\}$ .

**Metadata**: In the EH and OCRS matching methods, additional metadata is transmitted to the *evaluator* to analyze whether the extracted feature is suitable for the type of matching technique that is in use. This metadata is the sum of histogram bins in the EH method and the string length for the OCRS method.

**Evaluator**: Evaluates whether a match is found based on the similarity score, relative match confidence and other metadata extracted from the images. If a match is not found with a high confidence level, slide matching is performed more times using a technique that is different from the ones employed in earlier iterations. The evaluator takes into account the previously found distances. The evaluator outputs the best matched image to  $I$  in  $S$ . If no match is found then the best matched image is equal to NULL.

Target content-linking applications may require real-time processing or offline processing, which may also be a factor in the selection of the matching technique to employ. Color layout and edge matching techniques are more suitable for real-time implementations as they are less compute intensive than the OCR-String and Line Profile Matching techniques. In the following sections, we explain each of the matching methods in detail. Specific implementations of the matching algorithm are described in Section 5, where the experimental results are also presented.

### 3.1 Edge Histogram Matching

Text regions in images contain strong horizontal and vertical edges. Here, we employ local horizontal and vertical edge histograms to represent the amount of text and its layout in a slide image. An edge histogram is extracted as follows. First, a modified Sobel operator, shown in Figure 3 is applied to the image to obtain edge magnitudes. The parameter  $t$  estimates the width of character edges in a slide image. We use  $t=4$ , which works for the type of images, resolutions, and fonts that we capture. It is important to note that the performance of edge matching does not strongly depend on  $t$ , as long as the edges in all images are extracted using the same  $t$  value. After edge magnitudes are computed for an image, an edge is detected only if the edge magnitude is larger than a threshold and either an edge direction has changed or the distance between the location of the current pixel and the previous edge pixel is larger than  $t$ .

After horizontal and vertical edge pixels are extracted, the image is divided into  $N \times M$  segments and the number of horizontal and vertical edge pixels in each segment are stored in the horizontal,  $H_{hor}$ , and vertical,  $H_{ver}$ , edge histograms as follows:

$$H_{ver}(n, m) = \frac{1}{S_M S_N} \sum_{y=m}^{(m+1)S_M} \sum_{x=n}^{(n+1)S_N} \text{verticalEdge}[x][y],$$

where  $N$  is the number of horizontal segments and  $M$  is the number of vertical segments,  $\text{verticalEdge}[x][y]$  is the detected edge value at location  $(x, y)$ ,  $S_N$  and  $S_M$  are the height and width

of segment  $(n, m)$  in pixels and are computed by  $S_N = \text{ImageWidth}/N$  and  $S_M = \text{ImageHeight}/M$ . Horizontal edge histogram,  $H_{hor}$ , is found in a similar way.

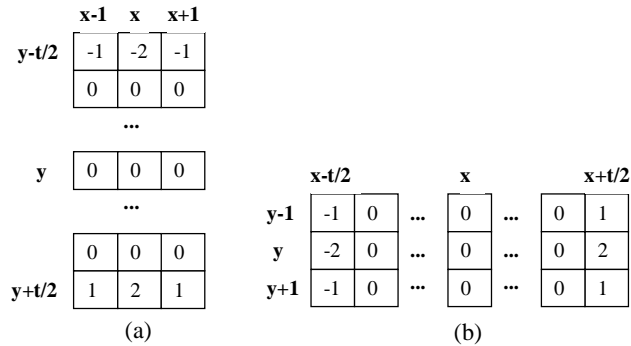


Figure 3. Modified Sobel operators used for (a) Horizontal (b) Vertical edge detection.

The resulting feature vector includes  $N \times M$  vertical edge histogram bins and  $N \times M$  horizontal edge histogram bins. Two global edge features are also included in the feature vector, which are computed by summing the values of the vertical and horizontal edge histograms separately and normalizing them by  $N \times M$  as follows.

$$G_{ver} = \frac{1}{MN} \sum_{m=0}^M \sum_{n=0}^N H_{ver}(n, m).$$

The resulting feature vector is as follows

$$\vec{F} = [G_{hor}, G_{ver}, H_{hor}(0,0), \dots, H_{hor}(n, m), H_{ver}(0,0), \dots, H_{ver}(n, m)]$$

The distance between the edge histogram feature vectors,  $\vec{F}_1$  and  $\vec{F}_2$ , are found by computing the L2 norm of their difference, which is the sum of squared differences of each vector value.

### 3.2 String Matching

Presentation slides generally contain text with oversized fonts in a color that contrasts with their background. Commercial OCR packages such as ScanSoft [13] and Transym [14] can be used to extract text from slide images. These packages usually correct for small skew and rotations, therefore we do not address these problems here.

After the images are OCR'd, string matching is performed on the text output to find a similarity score between two slide images. OCR results from slide images captured by different devices can vary widely. For example, the text output extracted from a digital camera image is generally less accurate than that obtained by OCR'ing the screen projection output for the same slide. In most applications one of the capture sources is likely to be more reliable than the other and the OCR results obtained from one of the sources could be close to the ground truth. We take this into

consideration when performing string matching and define the string that is obtained from the more reliable source as the ground truth string. The characters obtained for each slide are first concatenated. Then the similarity score between two strings are computed as follows.

$$d_s = (\ell_g - d_e) / \ell_g,$$

where  $\ell_g$  is the length of ground truth string and  $d_e$  is the edit distance between two strings. Edit distance is computed by counting the number of insertions and deletions required for matching. The punctuations and the extra characters in the string that is extracted from the less reliable source are ignored.

### 3.3 Line Profile Matching

Consider the video frame of a presentation slide shown in Figure 1.d. The low resolution and blur in the image makes obtaining accurate OCR results and thus performing string matching for this image very difficult. Edge matching cannot be successfully applied to this image either, as the slide region needs to be segmented and the blurred image can make segmentation challenging. We suggest matching of such images using line profile matching, which is performed as follows. First, the vertical edge detection method described in Section 3.1 is used to identify text regions, which have strong vertical edges. Edge are computed in each color space, i.e., R, G, B, and Luminance spaces. An edge pixel is detected if an edge pixel is identified in any of the color spaces. For each pixel location, a value,  $E_{xy}$ , is computed by accumulating the number of edge pixels in a neighborhood window  $K \times L$ . The pixels that have  $E_{xy}$  values that are larger than an adaptive threshold are marked as pixels that belong to a text region. Next, for each horizontal line in the image (if there are broken lines they can be connected), the maximum run of such pixels are computed to obtain a line profile. Line profiles of a slide captured by two different devices are represented in Figure 4.

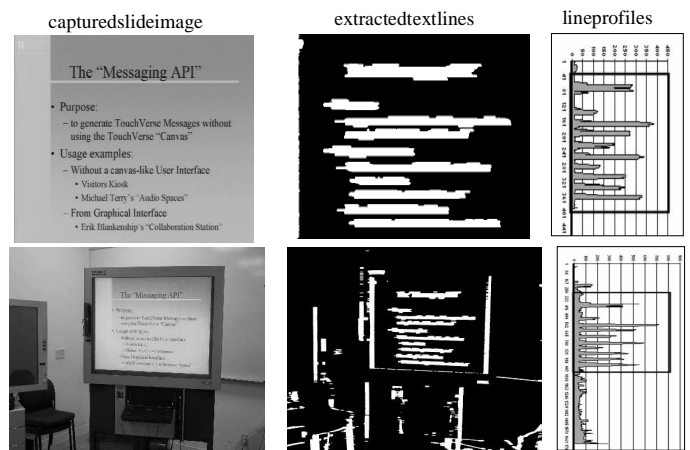


Figure 4. Line profiles of two captured slide images.

The peaks in the line profiles correspond to the text lines in the image. For matching line profiles of two images that can be of different resolutions, these values are first normalized. The

normalization is done by finding the peak value in the line profile (which is proportional to the longest text line in the image) and then scaling both the x and y values of the line profile with this peak value. This provides normalization in both horizontal and vertical directions. Then a feature vector is formed by the normalized line profile values.

The feature vectors of different images may be of different sizes. The distance between two feature vectors is found by aligning the two vectors with their peak value and computing the sum of absolute differences.

### 3.4 Layout Matching

There could be cases where a presentation slide does not contain any text or edges. In such cases the distance between two images can be found using color layout distance. Here, we down-sample each color channel of a slide image with an averaging filter to form a feature vector and compute the sum of absolute differences to find the layout distance between two vectors.

For grayscale and black and white images, the color layout matching technique uses only the luminance component of the image. Note that the slide region of an image needs to be segmented prior to computing the layout distance. If the image contains only the slide region, for example the images captured by the presentation recorder, then segmentation is not needed.

## 4. APPLICATIONS

Linking presentation streams and documents through slide image matching opens up possibilities for new ways to index, retrieve, and access meeting/presentation/lecture content and, consequently, for many interesting and useful applications. In the next sections we first give a brief overview of our presentation room setup and describe some new applications that are enabled by our slide matching algorithm.

Our presentation room is equipped with an omni-directional audiovisual meeting recorder and a presentation recorder. The meeting recorder captures 360-degree video at 30 frames per second and view selection is performed based on the sound directions [8]. The presentation recorder automatically captures what is displayed on the presentation screen/projector with the timestamps. To support a wide range of resolutions, the VGA output of the presenter's machine is connected to a scan converter where the VGA output is converted to an NTSC signal, captured by a frame grabber and saved in JPEG format at 640x480 resolution. The conversion from digital to analog and analog to digital results in some quality degradation in the image but this allows us to guarantee that the presentation recorder can capture video key frames from any laptop. The output of the meeting recorder and the presentation recorder are synchronized by time-stamps with post-hoc clock-skew correction.

### 4.1 Translating Projector

This application makes it possible for a presentation attendee to view the presentation slides in the language of his/her choice in

real-time. For example, during a lecture, a presenter can display his slides in English and the participants can use their Internet-enabled PDA's or laptops to view the presentation slides in German or Japanese using just a web browser. As the presenter changes his slides or goes back to some slides he already presented, the slides at the client's display gets updated accordingly in real-time. Neither presenter nor the viewer needs to install any special software on their computer.

Figure 5 presents how this application works. When the presentation slides are submitted to the server, the server extracts JPEG images of the slides with their page numbers and text from each presentation slide. Here, we use PowerPoint slides and the MS Office APIs to extract the JPEGs and the text from the symbolic presentation slides. Then, the server translates the slides using a commercial translation software package. The software package we use supports the translation of PowerPoint slides into 10 different languages while retaining the original formatting of the slides. The JPEG images of the translated slides are also saved using the MS Office APIs into individual directories. During the presentation, the presentation recorder captures screen images. When a new image is captured, the slide matching algorithm (using edge histograms, string matching, and layout matching) determines which of the PowerPoint slides the captured image matches. After identifying the PowerPoint file, the slide number, the web server updates the JPEG image viewed by the client according to the language of their choice and the currently displayed slide number. If the slide-matching algorithm fails to link the displayed slide to a PowerPoint slide, which could happen when the presenter is displaying something other than his slides on the screen, then the captured JPEG image is displayed at the client's viewer.

There could be alternative implementations for this application. For example, the JPEG images that are captured by the presentation recorder can be OCR'ed, translated and displayed at the client's machine. The downside of this implementation is that when erroneous OCR results are combined with the translation errors, the outcome may not be satisfactory. Also, it would be difficult to maintain the formatting and layout of presentation slides using this method.

Another alternative implementation is just transmitting the slide transitions to a server. In this case, the presenter needs to install a piece of software on her presentation machine that communicates the slide transitions to a server and needs to have a reliable network connection during the presentation. Because of these shortcomings, we did not consider these alternatives.

We successfully implemented the translating projector server in our presentation room. In some presentations, participants used their own laptops to access to the server and in some other presentations, where the audience speaks a common foreign language, we used a second projector to display the translated presentation slides in that language (usually Japanese). Even though in some cases there were errors in the automatic translation, the translated slides were well received and helped our visitors to better understand presentation's content.

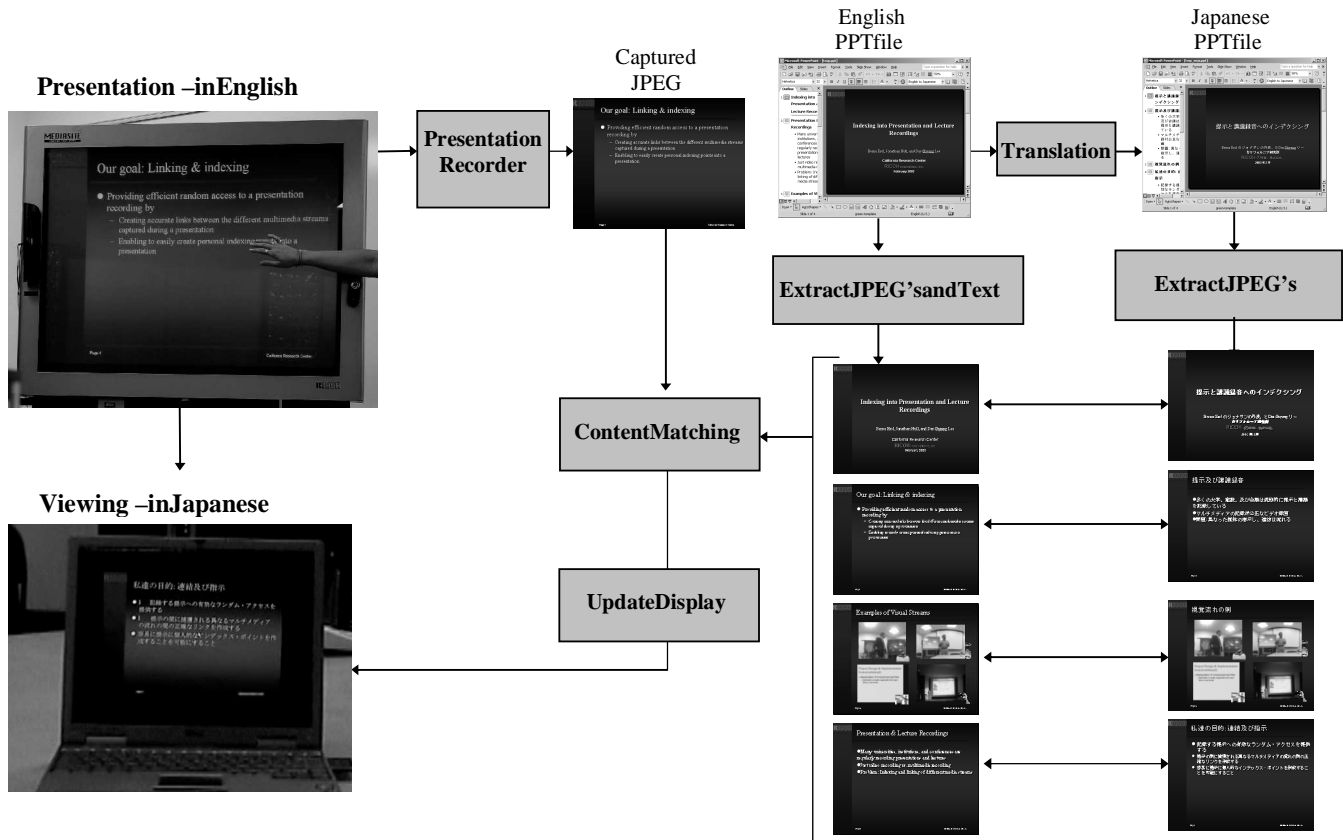


Figure 5. Multilingual presentations with Translating Projector-How it works.

## 4.2 Playback of Presentations from the Presentation Application

Presentation slides are generally prepared by an application (e.g., PowerPoint) that allows interaction with individual objects in a slide. The captured slide images on the other hand, are usually in a format that does not allow such an interaction (e.g., bitmap, JPEG, MPEG, etc). Therefore, where it is available, sharing and editing of the presentation slides are generally performed on the symbolic document. On the other hand, presentation slides generally do not contain much text and they are intended to be useful together with the presenter's explanations.

In this application, if a presentation was presented already, the audiovisual recording of the explanations and discussions related to each slide are automatically inserted into the symbolic presentation slides. Our matching algorithm (using edge histograms, string matching and layout matching) is employed to link the PowerPoint slides to the presentation recorder output and time-based synchronization is employed to link the presentation recorder output to the audiovisual recording. The audiovisual recordings as well as other relevant meeting recordings are inserted (as a link or embedded) into the symbolic document. This way, one can playback the audiovisual stream associated with the presentation directly from an application editing the presentation slides. This makes reviewing and sharing slides very efficient. For example, when a person needs

to edit or present some slides that were already presented by the original author, he can get more information on the presentation slides as he is editing without the original author's presence, he can listen to the questions asked about the presentation slides, and without the effort of finding and accessing to the relevant presentation recording.

## 4.3 Retrieval of Presentations with a Digital Camera Picture

As digital cameras are becoming widespread, it is becoming more common for attendees to take pictures of interesting slides in a presentation. More often than not, these images are useless because the context and the reason for taking them are forgotten later. In this application, users can submit presentation slide images as a query to a collection of presentation recordings and retrieve the audiovisual recording of the presenter talking about those particular slides. This is illustrated in Figure 6.

Most digital cameras attach time-stamps to the images they capture. If it is known which presentation recording is related to each captured image, then these time-stamps can be utilized to make a link between the digital camera image and the captured audiovisual presentation recording. However, in many cases it may not be practical for an attendee to keep track of the presentation sessions that he took each picture in, especially when more than one presentation session takes place in parallel.

In our implementation, we overcome this problem by retrieving images based on their slide content. The slide images captured by a digital camera are linked to the PowerPoint slides or screen images captured by our presentation recorder using our slide matching technique. A web based retrieval interface allows users to submit a digital camera image as a query and retrieve the relevant PowerPoint slide, screen capture, and the audiovisual recording captured at the time the slide was being presented. This way, a presentation attendee can easily refresh his/her memory about some particular slide or share them with others more effectively.

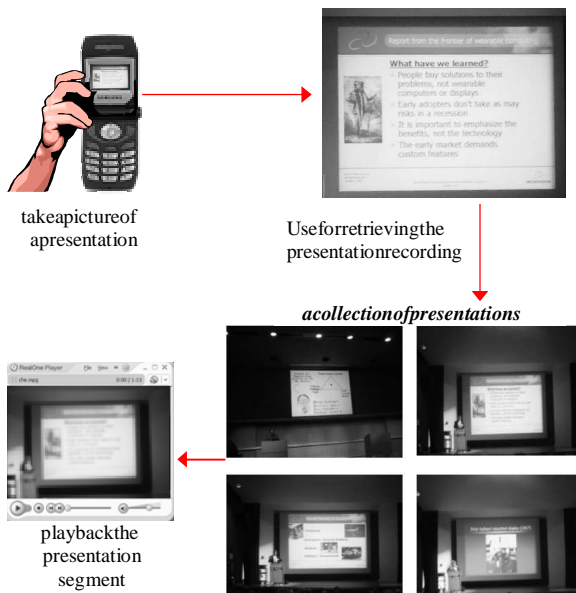


Figure 6. Indexing into presentations with a DC picture.

## 5. EXPERIMENTAL RESULTS

In this section, we compare the retrieval performance of our method to that of the DCT-based method presented in [12]. Both algorithms can match slides with images as well as text, and they do not restrict the type of slide background that one can use. The DCT-based method is based on down sampling the slide image to 64 × 64 pixels, performing a 64 point 2D-DCT, and then using the lowest frequency 256 coefficients to retrieve slide images. The performance of the algorithms is compared in the context of the applications described in Section 4.

We first present results for matching slide images obtained from the following two sources:

- Presentation recorder: The VGA output of presenter's computer is connected to a scan converter, captured at 640×480 resolution and saved as JPEG images.
- Presentation slides in PowerPoint that are saved as a series of JPEG images.

Presentation recorder output and PowerPoint images generally contain only slide regions, therefore we employed edge histogram, string and layout matching techniques for retrieval. These techniques are applied individually and the results are evaluated

for finding a matching slide with a high confidence. Pseudocode of this algorithm is given below.

```

Inputs:
I: Image to be matched
S: Image database

Output:
Match_I: Matched image with the highest confidence

Local variables, functions:
Cm: Match confidence value
Imatch 1, 2, 3: best matched images using individual techniques
MinDist: the distance of I to Imatch
NofEdges(I): Normalized average of edge histogram bins in I.
StringLength(I): Length of the extracted string from I.
T1, T2, T3, T4: thresholds*

edgeMatch(I, S, &Cm, &Imatch, &MinDist);
if (Cm > Th1 && MinDist < Th2 && NofEdges(I) > Th3)
    Match_I = Imatch;
else {
    StringMatch(I, S, &Cm, &Imatch2, &MinDist2);
    if (Imatch2 == Imatch && Cm > Th1/2 && StringLength(I) > Th4)
        Match_I = Imatch;
    else
        if (Cm > Th1 && MinDist2 < Th2 && StringLength(I) > Th4)
            Match_I = Imatch2;
    else {
        layoutMatch(I, S, &Cm, &Imatch3, &MinDist);
        if (Imatch3 == Imatch && Cm > Th1/2)
            Match_I = Imatch;
        else
            if (Cm > Th1 && MinDist < Th2)
                Match_I = Imatch3;
    }
}

```

\*The confidence threshold, Th<sub>1</sub> is 1.0. The accuracy of matching does not strongly depend on Th<sub>2</sub>, Th<sub>3</sub>. We selected Th<sub>2</sub> as 100, Th<sub>3</sub> as 0.01 and Th<sub>4</sub> as 20.

Our experimental setup was such that we presented 6 presentations with the number of slides totaling to 377. Some examples of presentation slides from our experimental setup are shown in Figure 7. It can be seen that the presentations we used in our experiments contain a variety of slides, e.g., light text on a dark background (1<sup>st</sup> and 6<sup>th</sup> presentations), dark text on a light foreground (2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup>, and 5<sup>th</sup> presentations), textured backgrounds (1<sup>st</sup>, 4<sup>th</sup>, and 6<sup>th</sup> presentations), various types of text layouts, slides composed of only text lines, only images, or a combination of images and text lines. In our experiment, all the presentations were captured with the presentation recorder and matched with the images and text obtained from the corresponding PowerPoint source files. The processing time of matching one image to a database of ~300 images was performed under 1 second on a 1GHz PC. Although OCR and string matching require high computing resources, they only slightly add to the computation time since they are performed only when a high confidence match (i.e.  $C_m < 1$ ) is not obtained by using only edge matching or when there are very few edges detected in an image. This was the case in less than 4% of the all matches. Also, layout matching is performed only when the length of the text extracted from a slide was insufficient ( $len < 20$ ) or the match confidence was low, which was the case in less than 1% of the all matches.

Retrieval rates for 6 presentations are represented in Table 2. Here, the retrieval rate (recall) is defined as the ratio of number of

correctly matched slides to the number of slides in a presentation. Note for each query slide image, there is only one ground truth image, therefore the number of ground truth items is equal to the number of slides in a presentation. As can be seen from the table, our method outperforms the DCT-based method, as it takes into account many properties of slide images. The DCT-based method particularly fails in cases for presentations, e.g., 1<sup>st</sup>, 4<sup>th</sup>, and 6<sup>th</sup> presentations, where the presentation slides contain a similar number of text lines and text fonts. This is mainly because the DCT-based method does not take into account the layout of a slide as the spatial information is lost in the frequency domain. Moreover, it also does not consider the semantic content. Our method, on the other hand, considers the semantic content via string matching for those images that have very similar text layouts.



Figure 7. Some slide examples from the presentations that are used in our experiments.

Note that because only edge matching is employed for the majority of the slides and OCR is avoided in most cases, our algorithm can be very fast. Therefore, it is suitable for real-time applications, particularly for the translating projector application.

Table 2. Slide matching results for different presentations.

Presentation	Number of slides	Retrieval rate - Proposed method	Retrieval rate - DCT based method
1	69	0.96	0.74
2	38	0.97	0.92
3	49	1.0	1.0
4	72	1.0	0.84
5	120	0.98	0.90
6	29	0.94	0.82
<b>ALL</b>	<b>377</b>	<b>0.98</b>	<b>0.87</b>

In the second set of experiments, we present the retrieval accuracy of presentation slides using digital camera pictures as queries. Matching is performed on images obtained from the following two sources:

- Presentation slides in PowerPoint that are saved as a series of JPEG images.
- Digital camera: 2-4 MPixel images mostly containing the slide region, with rotation less than  $\pm 5$  degrees.

In this experiment, for a given query, first the string matching technique is employed for retrieval. If the extracted string length is less than 20 characters or the match confidence is below a threshold  $Th_1$ , then retrieval is performed using the line profile

matching technique. Pseudo code of this algorithm is given below.

```

Inputs:
I: Image to be matched
S: Image database

Output:
Match_I: Matched image with the highest confidence

Local variables, functions:
Cm: Match confidence value
I_match 1,2,3: best matched images using individual techniques
MinDist: the distance of I to I_match.
StringLength(I): Length of the extracted string from I.
Th1=1.0, Th2=100, Th4=20

StringMatch(I, S, &Cm, &I_match, &MinDist);
if (Cm > Th1 && MinDist < Th2 && StringLength(I) > Th4)
    Match[I] = I_match;
else
    LineProfileMatch(I, S, &Cm, &I_match2, &MinDist);
    if (I_match2 == I_match && Cm > Th1/2)
        Match[I] = I_match;
    else
        if (Cm > Th1 && MinDist < Th2)
            Match[I] = I_match2;

```

Here, we performed experiments on 41 presentations with 1160 slides. Text extraction is performed on the PowerPoint slides using the Microsoft APIs at the time they are inserted into the database. The 109 digital camera images used as queries in our experiments were collected during 7 different presentations given in our lab. All the digital camera images contain at least one line of text. These images vary in terms of room lighting, use of flash, distance to the projector, motion blur, occlusion, etc. Some digital camera images taken during the presentations are shown in Figure 8. The process of matching one digital image to a database of 1160 images was performed in under 3 seconds on a 1GHz PC. Note that segmentation of the slide region from the digital camera image is not required for the matching technique employed here.



Figure 8. Examples of digital camera images of slides taken during presentations.

We also ran another set of experiments where the slide images captured by the *Presentation Recorder* are retrieved by using the digital camera images. In this case, the database contained 41 presentations with 4814 captured presentation recorder images (i.e. screen capture). Most of these images were presentation slides. Some of these images have the same slide content because a presenter may visit a slide more than once. For each query, the 10 slide images with the highest similarity score are retrieved.

The retrieval results obtained by querying the presentation slides with digital camera images are presented in Table 3. The retrieval rate is defined as the ratio of correctly retrieved relevant images to the number of relevant images in the database. The ground truth images are the presentation slides that match the digital camera images in terms of their slide content. As can be seen from Table 3, the average retrieval rate (recall) obtained by querying the PowerPoint slide database is 95%. The 3<sup>rd</sup> presentation, where the presentation slides generally contained very few lines of text, yielded the lowest retrieval rate. The average retrieval rate obtained using the Presentation Recorder database is 87%. Some of the presentations contained slides with small fonts (e.g., 2<sup>nd</sup> and 4<sup>th</sup> presentations). In these cases, the OCR on the slides captured by the Presentation Recorder yielded low accuracy, resulting in a lower retrieval performance. However, the retrieval performance using the PowerPoint slides was not affected by this since in this case we extract the text directly from the PowerPoint file instead of using OCR.

One should keep in mind that the overall retrieval accuracy of this method greatly depends on the quality (resolution, sharpness, etc.) of the digital camera image as much as the amount of text content in the captured presentation slide. Note that it is possible to improve the retrieval accuracy of slide image retrieval by incorporating information from the digital camera image timestamps.

**Table 3. Retrieval results when digital camera images are used to query presentations.**

Pres. no	Number of digital camera images	Ave Retrieval Rate (PowerPoint slides)	Ave Retrieval Rate (Presentation Recorder slides)
1	19	0.94	0.91
2	7	1.00	0.75
3	11	0.91	0.91
4	9	1.00	0.85
5	20	0.95	0.86
6	22	0.96	0.92
7	21	0.95	0.84
<b>ALL</b>	<b>109</b>	<b>0.95</b>	<b>0.87</b>

## 6. CONCLUSIONS AND FUTURE WORK

Presentation, lecture, and meeting recordings can be long and efficient access to these recordings is crucial to make them useful. Synchronizing multiple media streams captured during a presentation, as well as linking them with the relevant documents that are prepared before and after the presentations, such as symbolic presentation slides, meeting agenda, meeting minutes, etc. would potentially improve the usefulness of recorded presentations and make it easier to utilize this data in

our daily work and studies. In this paper, we addressed only a part of this critical issue and presented a method for content-based linking of presentation streams and documents. Some new applications and access techniques for multimedia recordings were also presented that potentially improve the efficiency of utilizing and retrieving captured multimedia presentations.

## 7. ACKNOWLEDGEMENTS

We would like to thank Jamey Graham for his support and advice, and Daniel Van Olst, Kim McCall, and Bradley Rhodes, for helping to collect test images.

## 8. REFERENCES

- [1] Pimentel, M.G.C, Abowd, G., Ishiguro, Y., "Linking by Interacting: a Paradigm for Authoring Hypertext" Proceedings of ACM Hypertext 2000, pp.39-48, May, 2000.
- [2] Mukhopadhyay, S., and Smith, B., "Passive capture and structuring of lectures", ACM Multimedia pp. 477-487, 1999.
- [3] Jason, A., Brotherton, Bhalodia, J.R., and Abowd, Gregory G. D., "Automated Capture, Integration, and Visualization of Multiple Media Streams", Proceedings of IEEE Multimedia'98, pp.54-63, 1998.
- [4] Presenter Inc, www.presenter.com
- [5] Müller, R., and Ottmann, T., "The "Authoring of the Fly" System for Automated Recording and Replay of (Tele)presentations", ACM/Springer Multimedia Systems Journal, Vol.8, No.3, pp.158-176, 2000.
- [6] Chiu, P., Kapuskar, A., Reitmeier, S., and Wilcox, L., "Room with a Rear View: Meeting Capture in a Multimedia Conference Room", IEEE Multimedia Magazine, pp.48-54, vol.7, no.4, Oct-Dec 2000.
- [7] Multi-university Research Laboratory, murl.microsoft.com
- [8] Lee, D.S, Erol, B., Graham, J., Hull, J.J., and Murata, N., "Portable Meeting Recorder", ACM Multimedia Conference, pp.493-502, 2002.
- [9] Stifelman, L., "The Audio Notebook: Paper and Pen Interaction with Structured Speech," PhD Thesis, MIT 1997.
- [10] Chiu, P., Kapuskar, A., Wilcox, L., "Meeting Capture in a Media Enriched Conference Room," 2nd International Workshop on Cooperative Buildings, pp.79-88, 1999.
- [11] Franklin, D., and Bradshaw, S., and Hammond, K.J., "Jabberwocky: you don't have to be a rocket scientist to change slides for a hydrogen combustion lecture", Intelligent User Interfaces, pp.98-105, 2000.
- [12] Chiu, P., Foote, J., Girgensohn, A., and Boreczky, J., "Automatically Linking Multimedia Meeting Documents by Image Matching" proceedings of Hypertext'00, ACM Press, pp.244-245, 2000.
- [13] Scansoft Capture Development System, www.scansoft.com
- [14] Transym OCR Engine, http://www.transym.com/