Matching Slides To Presentation Videos Using SIFT and Scene Background Matching

Quanfu Fan
Department of Computer Science
University of Arizona, Tucson, AZ85721
quanfu@cs.arizona.edu

Kobus Barnard
Department of Computer Science
University of Arizona, Tucson, AZ85721
kobus@cs.arizona.edu

Arnon Amir
IBM Almaden Research Center
650 Harry Road, San Jose, CA95120
arnon@almaden.ibm.com

Alon Efrat
Department of Computer Science
University of Arizona, Tucson, AZ85721
alon@cs.arizona.edu

Ming Lin
Department of Management Information Systems
University of Arizona, Tucson, AZ85721
mlin@email.arizona.edu

ABSTRACT

We present a general approach for automatically matching electronic slides to videos of corresponding presentations for use in distance learning and video proceedings of conferences. We deal with a large variety of videos, various frame compositions and color balances, arbitrary slides sequence and with dynamic cameras switching, pan, tilt and zoom. To achieve high accuracy, we develop a two-phases process with unsupervised scene background modelling. In the first phase, scale invariant feature transform (SIFT) keypoints are applied to frame to slide matching, under constraint projective transformation (constraint homography) using a random sample consensus (RANSAC). Successful first-phase matches are then used to automatically build a scene background model. In the second phase the background model is applied to the remaining unmatched frames to boost the matching performance for difficult cases such as wide field of view camera shots where the slide shows as a small portion of the frame. We also show that color correction is helpful when color-related similarity measures are used for identifying slides. We provide detailed quantitative experimentation results characterizing the effect of each part of our approach. The results show that our approach is robust and achieves high performance on matching slides to a number of videos with different styles.

Categories and Subject Descriptors

I.2.10 [**Information Systems**]: Information Storage and Retrieval

General Terms

Algorithms, Performance, Experimentation

Keywords

Distance Learning, Slides, Presentation Videos, SIFT Keypoints, RANSAC, Video Indexing, Color Correction

1. INTRODUCTION

In this work we consider matching electronic slides to presentation videos captured by one or several cameras, either fixed or allowed to pan tilt and zoom. More formally, given a sequence of video frames $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$ and images of the electronic slides $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$ associated with them, we are interested in finding a mapping function $\mathcal{M} : \mathcal{F} \rightarrow \mathcal{S}$ such that $\mathcal{M}(f_i) = s_j$ if frame f_i contains slide s_j and $\mathcal{M}(f_i) = 0$ when there is no slide visible in frame f_i.

Matching slides to videos provides an attractive way of indexing videos by slides for searching and browsing. By finding the original electronic slide that is displayed at any point in time along the video, it is possible to display clear, high resolution slides side by side with the video. This matching automates a currently manual process in the preparation of class content for distance learning, therefore reducing time to publish and labor costs. It further allows to adjust the quality of the slide images in video frames (resolution, color, and contrast), and to index and retrieve video segments using the textual content of the corresponding electronic slides.

Slides to video matching has been studied for about a decade. Early work such as the Classroom 2000 Project [1] and BMRC Lecture Browser [14] manually edit time stamps to match the slides to the video clips. Recently some automatic approaches [3, 5, 6, 10, 12, 15] have been proposed to match or synchronize slides to videos. Such approaches usually involve two steps: 1) locating slides or extracting texts in the video frames; and 2) identifying slides by using some recognition methods such as template matching or string matching. In such a matching process, the first step is crucial as it directly determines the performance of the

MIR'06, October 26–27, 2006, Santa Barbara, California, USA.

follow-up recognition task.

In what follows, a *slide* refers to an image, automatically extracted from a presentation file (e.g., PPT or PRZ files). Video frames, often denoted *keyframes* when they represent segments of the video, or in short *frames*, are extracted from the (compressed) digital video of the presentation.

We divide the video frames into 3 types, or categories. A frame is called a *full-slide* frame if the entire frame shows the (usually entire) slide content. Otherwise, it is called a *small-slide* frame if it contains both a slide area and a substantial portion of scene background. These are usually wide field of view shots of the presenter along with the projection screen. A frame without any slide is referred to as a *no-slide* frame.

In a video capturing system, where several cameras are mixed and are allowed to pan tilt and zoom, the contents of the video frames can vary greatly. For example, when a slide is captured in the video, it may appear small, full-frame, or clipped (camera zoom-in). Further, the slide content may appear in whole or in part, such as during an animation, and might suffer partial occlusion, e.g., by the speaker. Even worse, not only the geometry and content of the projected slide varies, even the colors might change greatly due to switching between cameras with different settings, and any dynamic changes in camera settings, e.g. an automatic shutter response to changes in ambient illumination or slide brightness. When no electronic slide is present, the frame may show the speaker and some *scene background*, or just *background* (e.g., part of the classroom, the audience, or both that is visible in some video frames). In other cases the frame may still show the projector screen, displaying non-slide content such as a demo, a video, or a web page. Throughout the paper we refer by background to the classroom scene background (not to be confused with the slides' template or "background", that are not discussed in this work).

Figure 1 shows different types of frames captured in a presentation. In many cases the slide's text is not recognizable, so text-based matching approaches are not appropriate. Also, when large amount of camera zoom is applied it might become very challenging to accurately spot the corresponding slide area. It is therefore desired to have a unifying matching approach that can handle all these difficulties without being limited by the camera setup or the frame type.

Towards this goal, we propose a new robust approach to match slides to video frames regardless of the frame type. Our algorithm uses random sample consensus (RANSAC) [7] to robustly estimate homographies between frames and slides based on scale invariant feature transform (SIFT) keypoint matching [11]. SIFT keypoint features are highly distinctive and are invariant to image scale and rotation. They can provide correct matching in images subject to noise, blurring and illumination changes. Hence our approach can deal with a large varieties of videos, including those of less than professional quality.

As one may assume, full-frame slides are the easiest to match. When slides are shown small in the video, the matching task is harder, and the RANSAC process might fail to find a good slide match. We propose several strategies to help RANSAC deal with this difficulty effectively and efficiently. Our approach is to process the video frames in phases, where in the first phase we fetch to find high-confidence matches for the "easier" cases and the rest, unmatched slides, are being classified by their types. The successful first-phase matches are then used to build a scene background model automatically. In the second phase, the model is applied to the remaining unmatched frames to boost the matching performance of the more difficult cases. We also show that color correction is helpful when color-related similarity measures are used for identifying slides.

The remainder of the paper is organized as follows. Section 2 gives a short review on SIFT features, homography and RANSAC. Section 3 presents our slides to frame matching algorithm in details. Section 4 describe a simple color correction model. Section 5 demonstrates the experimental results. Finally, Section 6 concludes our results and outlines future work.

2. OVERVIEW: SIFT KEYPOINTS, HOMOGRAPHIES AND RANSAC

The basic step in matching a video frame with its corresponding slide involves with estimating the geometric transformation between the frame and the slide, identifying the corresponding slide (i.e., slide number), and computing the level of confidence in this match. In oppose to most past approaches which aim at separating between the transformation estimation and the slide identification by first looking for the slide region in the frame, here we simultaneously solve for both the transformation and the identification problems using RANSAC on SIFT keypoints, and further use those to estimate the matching confidence. Our approach is fully automatic and can handle video capture systems with few limits on the motions (pan, tilt, zoom and even movement) of the cameras.

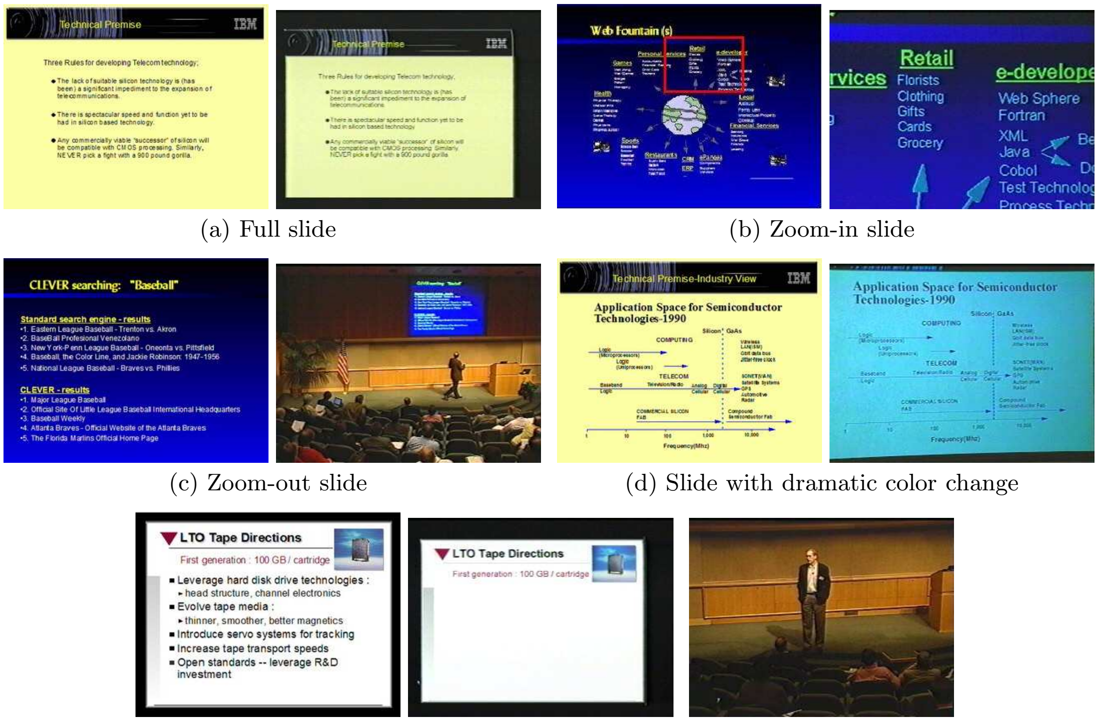
2.1 The SIFT Descriptor

Recently, great progress has been made in object recognition by using local descriptors such as the scale invariant feature transform (SIFT) keypoints [11] used here. SIFT keypoints are points of local gray-levels maxima and minima, detected in a set of difference-of-Gaussian images in the scale space. Each keypoint is associated with a location, scale, orientation and a descriptor - a 128-dimensional feature vector that captures the statistics of gradient orientations around the keypoint. SIFT keypoints are scale and rotation invariant and have been shown robust to illumination change and viewpoint change. Figure 2 shows the SIFT keypoints detected in a slide. Since SIFT is based on the local gradient distribution, as seen in the figure, heavily textured regions produce more keypoints than a color-homogeneous region. Fortunately for our application, text on slides yields many, well distinct keypoints.

Given the keypoints detected in two images A and B, Lowe [11] presents a simple matching scheme based on the saliency of the keypoints. A keypoint P_A from image A is considered a match to a keypoint P_B from image B if P_A is the nearest neighbor of P_B in the descriptor's feature space and

$$\frac{d(P_A, P_B)^2}{d(P'_A, P_B)^2} < \tau^2 \tag{1}$$

where $d(.,.)$ denotes the Euclidean distance between the descriptors of the two keypoints and P'_A is the second nearest keypoint of P_B in image A. For simplicity, we refer to this matching algorithm as the *nearest neighbor (NN)* algorithm.


Technical Premise
IBM
Three Rules for developing Telecom technology;
The lack of suitable silicon technology is (has been) a significant impediment to the expansion of telecommunications.
There is spectacular speed and function yet to be had in silicon based technology.
Any commercially viable "successor" of silicon will be compatible with CMOS processing. Similarly, NEVER pick a fight with a 900 pound gorilla.
Web Fountain (s)
Retail
Florists
Clothing
Gifts
Cards
Grocery
e-develope
Web Sphere
Fortran
XML
Java
Cobol
Test Technolog
CLEVER searching: "Baseball"
Standard search engine - results
•1. Eastern League Baseball - Trenton vs. Akron
•2. BaseBall Profesional Venezolano
•3. New York-Penn League Baseball - Oneonta vs. Pittsfield
•4. Baseball, the Color Line, and Jackie Robinson: 1947-1956
•5. National League Baseball - Braves vs. Phillies
CLEVER - results
•1. Major League Baseball
•2. Official Site Of Little League Baseball International Headquarters
•3. Baseball Weekly
•4. Atlanta Braves - Official Website of the Atlanta Braves
•5. The Florida Marlins Official Home Page
Technical Premise-Industry View
Application Space for Semiconductor Technologies-1990
COMPUTING
TELECOM
COMMERCIAL SILICON FAB
Compound Semiconductor Fab
Frequency(Mhz)
LTO Tape Directions
First generation : 100 GB / cartridge
Leverage hard disk drive technologies :
head structure, channel electronics
Evolve tape media :
thinner, smoother, better magnetics
Introduce servo systems for tracking
Increase tape transport speeds
Open standards -- leverage R&D investment

(a) Full slide

(b) Zoom-in slide

(c) Zoom-out slide

(d) Slide with dramatic color change

(e) Animated slide

(f) No slide

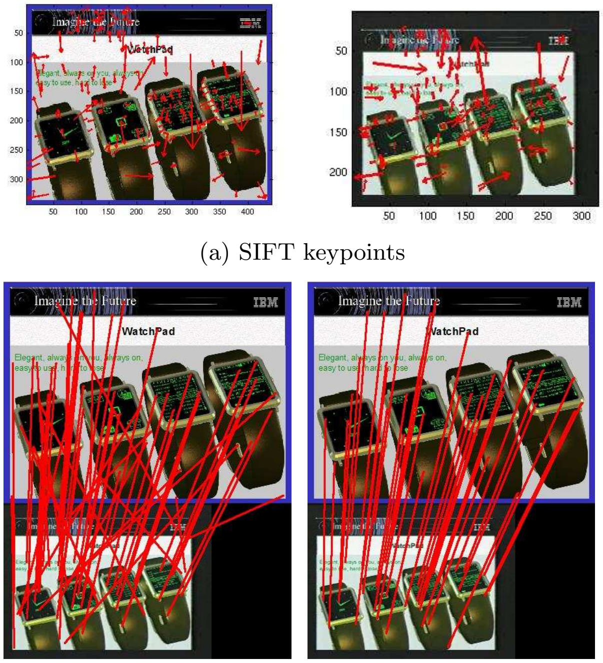
Figure 1: Different frame types captured by camera. Except 1(f), which shows two frames without slide, each pair includes an original slide image (left) and one of the sample video frames of the slide (right). In (b), the red box in the left image indicates the original slide area of the frame in the right. According to the definitions in Section 1, (a), (b), (d) and (e) are full-slide frames, (c) is small-slide frame and (f) is no-slide frame.

In [11], a threshold $\tau = 0.8$ was selected for object recognition. In our experiments, we found that this threshold excludes the majority of outliers while keeping a good portion of the correct matches. Hence, we set $\tau = 0.8$ in all our experiments. Fig 2 shows a putative matching result found by this scheme.

The above matching scheme searches keypoint matches in the whole images. However, if the transformation between two images is given, we can project the keypoints in one image to the other one and find keypoint matches locally within some range r, i.e we add another criterion as follows,

$$|(P_B(A)P_B)| \leq r \tag{2}$$

where $|.|$ is the Euclidean distance between P_B and the projection of P_A in the image B. We refer this searching scheme as **local mode** in comparison to the above **global mode**. When slides are small, keypoints on them become less distinctive and a global search rejects many correct matches. Instead, a local search can not only gives more keypoint correspondences, but also be more likely to ensure more correct ones due to geometric constraints (the known transformation). In the following section, we will discuss how to find the transformation between a slide and a frame to make this scheme applicable.



(a) SIFT keypoints

(b) Putative matching (c) Correct matching

Figure 2: Keypoint matching. The top two images shows keypoints detected in two images. An arrow attached to each keypoint shows the associated scale and rotation features. The image on the bottom left shows matches proposed by simple nearest neighbor algorithm. The image on the right shows proper matches that share a homography from slide to frame. For clarity we only display about one quarter of the keypoint matches.

2.2 Fitting a Homography using RANSAC

Outliers Ratio (%)	10%	25%	50%	75%
# RANSAC iterations	4	12	71	1177

Table 1: The number of RANSAC iterations required to ensure 99% confidence that at least one sample will have no outliers for a sample size of 4 keypoints (homography).

The mapping between a coplanar set of points and its perspective projection on an image plane is provided by a *Homography*. The homography H between a slide and its projected image in the frame plane can be determined by four or more pairs of corresponding keypoints by solving $X' = HX$ where X is a set of slide keypoints and X' is the corresponding frame keypoints. In this work, we used the Normalized Direct Linear Transformation (See [9] for details.) to estimate H.

SIFT keypoints are highly distinctive. As shown in Fig 2, the simple matching scheme can give a set of putative correspondences with a good portion of correct keypoint matches. However, the outlying correspondences, even a few, can severely affect the estimation of a homography. Here we use RANSAC to search for the true keypoint correspondences by imposing a homography on the putative correspondences found by the NN algorithm.

RANSAC is an iterative random algorithm. At each iteration, a randomly selected subset of four pairs of matched keypoints is used to compute a hypothesized homography. The hypothesis is then evaluated by checking how many of the remaining matched pairs of keypoints are consistent with it. Hence the required number of iterations to ensure high probability of detection depends on the percentage of the outliers in the data. Table 1 shows that the number of iterations required for homography estimation increases dramatically as the rate of outliers increases. In our experimentations, when the slide area occupies all or most of the frame, there are less than 50% outliers in the putative metched keypoints. In such a case, testing 100 hypotheses is sufficient to ensure a 99% chance of finding the correct homography H. In Section 3, we address some of the difficult cases where more samplings are required.

3. THE SLIDES TO VIDEO MATCHING ALGORITHM

In general, RANSAC works very well on our test data and achieves impressively high performance. However, it faces a few challenges due to the high complexity of this data. Firstly, when the slide's area is very small in the frame and the slide does NOT have rich texture (e.g., slides containing only a single plot, a small table or very little text) RANSAC might fail to find the homography due to the interference of outliers from the background scene surrounding the slide area in the frame. Secondly, more than a third of the frames in our data has no slides. Matching them to slides takes a lot of time. We propose an approach called *background matching* to overcome these problems.

Our algorithm includes three stages - two RANSAC-based recognition phases with an unsupervised scene background modelling in between. Initially, all frames are marked *UNDECIDED*, i.e. their types are unknown. RANSAC is then run with a small number of iterations to gather useful information about the frame types. In the second stage *background matching* and a binary classifier are applied to further determine the frame types for remaining undecided ones. Finally, RANSAC is run again, but only on the unsuccessful frames with slides or the undecided slides from the first run. Note that *FULL*, *SMALL* and *NOSLIDE* in the following algorithm correspond to full-slide , small-slide and no-slide frames defined above, respectively.

The algorithm is summarized as follows:,

1 Mark all frames as *UNDECIDED*.

2 Run RANSAC with a number of iterations N_1 to find slide matches for each frame. If an acceptable slide match for a frame is found, mark the frame either as *FULL* or as *SMALL* according to the ratio of the slide area over the whole image.

3 Using the information obtained in Step 2, build a binary classifier to detect all full-slide frames and mark them as *FULL*.

4 If no *SMALL* frames were found in Step 2, skip to step 5. Otherwise, create an (unsupervised) scene background model, apply background matching on all *UNDECIDED* frames and classify them as *SMALL* or *NOSLIDE*.

5 Run RANSAC again with a number of iterations N_2 on all frames that have not successfully claimed slide matches in the first run and are not labelled *NOSLIDE*. For *SMALL* frames, use for matching only keypoints from the slide area.

In our experiments, we set $N1 = 100$ and $N2 = 400$ based on Table 1.

3.1 Matching Score

3.1.1 Keypoints-based Score

Each frame f_j is compared to all the slides to find the best matching slide. Let $B(s_i|f_j)$ denote the quality of matching between slide s_i and frame f_j. It can be regarded as the similarity between s_i and f_j. Let k_{ij} be the number of keypoint correspondences between s_i and f_j that are consistent with the best homography found by the RANSAC between s_i and f_j, then a simple expression of $B(s_i|f_j)$ is,

$$B(s_i|f_j) = \begin{cases} \frac{k_{ij}}{\sum_i k_{ij}} & \text{if } k_{ij} \geq m \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

That is, the best matching slide is accepted if and only if the score passes a certain threshold, m. We experiment with different values of m, whereas higher value provides higher confidence is accepted matches with the tradeoff of higher risk to reject correct matches with small number of correspondences. The effect of this threshold on the number of errors is evaluated in Section 5.

3.1.2 Normalized Cross Correlation (NCC) Score

The normalized cross correlation (NCC) preferred in template matching [8] is another important similarity measure between two images. We define another similarity score as

$$B(s_i|f_j) = \begin{cases} C\rho_{ij} & \text{if } k_{ij} \geq m \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where ρ_{ij} is the NCC between the projected image of s_i in f_j (after color correction) and the slide content of f_j. C is a constant that makes B a valid probability. ρ is set to 0 if it is less than 0. m is the same threshold as that defined in Equation 3.

3.2 Scene Background Matching

The scene background interference with the RANSAC algorithm is in large eliminated by using the background matching. If the slide area of a small-slide is known, we can reduce the background outliers affect on RANSAC by pruning those erroneous keypoint matches from the area surrounding the detected slide area. Although approaches proposed in previous work such as [6] and [10] could be used to detect the slide area of in the video, they are not robust when the color of slide area is not very distinguishable from to the scene in the frames. We present a new approach to detect the slide area of a small-slide frame by automatically matching the background between frames.

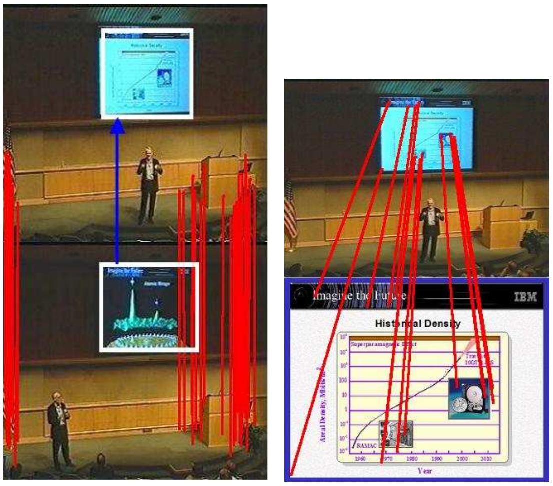
As can be observed, small-slide frames usually share some static objects in the scene such as floor, podium, walls, audience etc (see Fig 1(c)). Let f_1 be a small-slide frame with known slide area detected by RANSAC. We call f_1 a *reference frame*. Let f_2 be another small-slide frame with unknown slide area. Similar to matching a slide to a frame, we match between frame f_2 and frame f_1 by RANSAC and estimate the transformation H between them. The known area of f_1 can now be transformed to f_2 by H, to produce the slide area in f_2. Since the transformation H is established by matching the shared background objects between the two frames, we call this method "*background matching*". Note that the background is not a planar object, so the assumption for the simple homography does not hold. A more well-grounded transformation in this case is the fundamental matrix [9] between two frames. However, as long as the parallax between the frames is minimal (that is, camera may undergo pan, tilt and zoom but no significant translation), homography is still an acceptable approximation.

Usually we only need to know one reference frame, which can be obtained from the first RANSAC run. If more than one reference frame is available, we can combine them together to get a "big" reference frame for efficiency. We briefly describe this idea as follows. Let $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ be a set of small-slide frames with known slide areas $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$. We pick the frame f_k that has a slide located closest to the the image center and match all the other reference frames to f_k. For each frame $f_i (i \neq k)$, we transform all its keypoints and its slide area r_i to the coordinates of f_k. Let $K_{i'}$ and $r_{i'}$ be the new keypoint set and the new slide area for f_i, respectively. Then, the new reference frame can be expressed as $f_{new} = (\bigcup_{i=1}^m K_{i'}, \bigcup_{i=1}^m i^{'})$. Redundant or similar keypoints are removed in the united set.

Once the slide area is spotted in a frame, we can prune the erroneous keypoint matches from the background and apply the local mode search to find more correct keypoint matches. Thus, we greatly increase our chance of detecting the correct slide match to the frame in the second RANSAC run. Fig 3 shows the background matching between two frames and the successfully identified slide area by this approach.

3.3 Detecting No-slide Frames by SVM

Matching frames without slide to slides spends a lot of time and can introduce false matches. Thus we would like to



(a) Scene background matching (b) Keypoint matching inside the slide area only

Figure 3: Scene background matching between two video frames. First, background matching is applied between two small-slide frames (left). The white box in the bottom frame bounds the slide area, which is known from a successful match in the first RANSAC phase. It is used to infer the slide area in the top frame. Next (right), keypoint matching is applied between the projected slide area and the slide, eliminating nearly all scene background keypoints.

prune them from the process as soon as possible. However, without prior knowledge, detection of background and no-slide frames is not trivial. In this section, we show that it is possible to detect no-slide frames without any prior knowledge about the video.

Usually, there is significant visual difference between full-slide frames and other frames containing substantial background (i.e small-slide and no-slide frames). We can first separate full-slide frames from others by using a binary classifier. For frames not classified as full-slide frame, the background matching described above can further tell whether they are a small-slide or no-slide frame.

We use linear-kernel SVM [4] as our binary classifier for this task. The image features used are Color Coherence Vector (CCV), a color histogram that incorporates spatial information [13]. Given two images I and I', their CCV distance is defined in [13] as,

$$d_G(I, I') = \sum_{i=1}^{n} |\alpha_i - \alpha'_i| + |\beta_i - \beta'_i| \tag{5}$$

where $G_I = \langle(\alpha_1, \beta_1), \ldots, (\alpha_n, \beta_n)\rangle$ and $G'_I = \langle(\alpha'_1, \beta'_1), \ldots, (\alpha'_n, \beta'_n)\rangle$ are the normalized CCVs of I and I', respectively.

We construct the SVM training data set as follows. Let $\mathcal{F}_l$ be the set of full-slide frames detected in the first RANSAC run in the matching algorithm. Similarly, $\mathcal{F}_s$ is defined as the set of small-slide frames detected. We set the positive samples $\mathcal{D}^+ = \mathcal{S} \cup \mathcal{F}_l$ where $\mathcal{S}$ is the set of original slides. The reason for including $\mathcal{F}_l$ in $\mathcal{D}^+$ is to count for the possible varied light conditions. The negative samples $\mathcal{D}^-$ are selected as $\mathcal{D}^- = \mathcal{F}_s \cup \mathcal{B}$ where B is the k farthest frames away from $\mathcal{D}^+$ in feature space. The distance of a frame f_i to $\mathcal{D}^+$ is formally defined as,

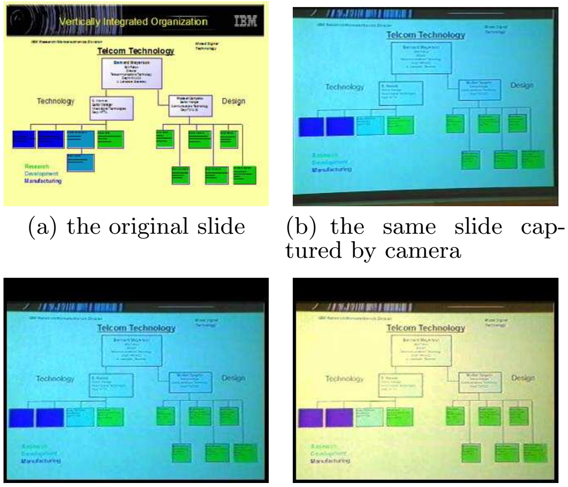
$$d_{D^+}(f_i) = \min_{f_j \in \mathcal{D}^+} d_G(I_{f_i}, I_{f_j}) \tag{6}$$

k is determined based on the size of $\mathcal{D}^+$ and $\mathcal{F}_s$.

One drawback of this approach is that it has to rely on background matching to single out the small-slide frames first. If the first RANSAC run is unable to provide any information for background matching, this approach can only separate full-slide frames from others.

4. COLOR CORRECTION

Images can vary greatly in color if captured under different lighting conditions, as shown in Fig 1(d). If an algorithm relies on a color-related measure to identify slides in videos, then it has to take color issue into accounted. However, most of the previous work have not addressed this issue.



(a) the original slide (b) the same slide captured by camera

(c) the registered slide (d) the slide after color correction

Figure 4: Color correction

One simple color constancy model is a single linear transformation. Let $C_s = (R_s, G_s, B_s)$ be the color of a pixel in the original slide image and $C_f = (R_f, G_f, B_f)$ be the color of a pixel in the image registered from a frame to the slide image. We can map C_f to C_s by,

$$C_f = MC_s \tag{7}$$

Solving (7) yields $M = C_f C_s^T (C_s C_s^T)^{-1}$ and the corrected color C'_s can be obtained by,

$$C'_s = M^{-1} C_f \tag{8}$$

Figure 4 shows a slide image after color correction.

5. EXPERIMENTAL RESULTS

In this section, we conduct detailed performance analysis on each part of our algorithms. We first look at the overall accuracy of our algorithms on recognizing slides in videos. We then examine in detail the effectiveness of the background matching method.

Dataset	Video	Duration (min)	Full Slides	Small Slides	No Slides	Total	PPT Slides
CONF1	#1	47	33	9	61	103	29
	#2	55	76	3	72	151	39
	#3	41	38	6	53	97	27
	#4	20	20	8	23	51	21
	#5	39	41	12	64	117	34
	#6	49	53	42	59	154	67
CONF2	#1	68	122	3	103	228	63
	#2	54	58	1	104	163	68
	#3	63	50	0	90	140	49
	#4	52	40	1	61	102	33
	#5	47	17	0	52	69	53
UNIV	#1	39	33	9	61	103	44
	#2	48	76	3	72	151	48

Table 2: Summary of the video data used in our experiments.

5.1 Video Data

We construct a set of 13 presentation pairs (MPEG video + presentation file); 6 presentations from a corporate conference and 5 presentations from a scientific conference, both captured using a similar setup of three pan-tilt-zoom cameras, with live video editing; one camera tracks the speaker, one camera covers the projector screen and is used to zoom in on the slides, and the third camera captures the audience [2]. Two more presentations are university seminars (denoted UNIV) captured by two cameras (one full-slide and the other gives either small-slide or audience views). All presentation files were prepared and delivered by different speakers, thus in the corporate conference all speakers used the same slides template. The video data is summarized in Table 2.

We manually constructed a ground truth matching between frames and slides for evaluation purposes. Each keyframe containing a slide is labelled with the slide number and with full-slide or small-slide . Keyframes containing no slides are marked with 0. The few frames showing missing slides are marked as “missing” and are not considered in the evaluation.

5.2 Video And Slide Image Processing

The videos were first processed by shot boundary detection and one keyframe was extracted from each shot. The frame size is 320×240 for all the videos and the size of slides is 443×342. We resized the slides to 320×247 for efficiency considerations. A few PPT slides were missing in videos 3 and 4 because they were removed from the PPT files by the speakers after the talk and before providing us with their files.

The precision measure P used in our experiments is defined as,

$$P = 1 - \frac{\#\text{ of correctly identified frames}}{\#\text{ of ground truth frames}} \quad (9)$$

Throughout the results section we look at missrecognition error counts, marked by the number of misrecongnized/total slides. A slide is considered correctly recognized if it is matched by the system to the same PPT slide number as marked in the ground truth. Hence an error counts as either matching with a wrong slide (rare) or failure to match with any slide (the more common failure mode).

We experimented with two image similarity measures: one is the counts of keypoint matches (KP), described in Section 3.1.1 and the other is the normalized cross correlation (**NCC**) [8] defined in Section 3.1.2. The results on the first data set are presented in Table 3, showing clear advantage for using color correction with the NCC measure on full-slide frames. However, Keypoint-based score outperforms NCC and requires no color correction, so we selected it as our similarity measure for the rest of the experimentation.

We considered the first phase algorithm with fixed 100 RANSAC iterations as the base algorithm (**keypoint matching**). Table 4 display the results. It can be seen that baseline recognition results are more than 97% accurate for full-slide frames and even more accurate in classification of no-slide frames. Most errors occur, as expected, in small-slide frames.

It is worth to note that we experimented with keypoint mapping either from frame to slide or from slide to frame. Since the frame and the slide produce substantially different sets of keypoints, and we use nearest neighbors to find the matches, this mapping is not a symmetric relationship. We found a clear advantage to mapping frame points to slide points over going the other direction, in particular for small-slide frames. This is partly because small-slide frames have much fewer keypoints in the slide area, and mapping those keypoints onto the slide has much higher chance to find correct correspondences than going the other direction. Hence all the experimentation were carried this way.

Next we compare the **background matching** algorithm described in Section 3.2 with the **keypoint matching** performance. The **background matching** algorithm uses a fixed number of 400 iteration in the second phase RANSAC run. Table 5 shows the performance of these two algorithms on the data set. In this table, KP(100) denotes the **keypoint matching** performance and KP(400) denotes the performance of the same baseline algorithm when the RANSAC is allowed for 400 iterations. The background matching, denoted as BP(), was tested twice, once in local-mode matching and second in global-mode matching. Each mode uses 100 RANSAC iterations in the first phase and 400 RANSAC iterations in the second phase, after background matching.

As we can see from Table 5, both **keypoint matching** and **background matching** achieve high recognition performance, with **background matching** performing significantly better than **keypoint matching** on small slides. The improvement is in part attributed to running 400 more RANSAC iterations - hence we provide the results of the baseline KP(400) for comparison. Note, however, that this run is much slower than the other runs because it runs 400 iterations on all the frames, as oppose to running those only on the small slides in the background matching method. Moreover, the background matching method still outperforms the KP(400) run.

There is a noticeable difference between the full-slide recognition error on CONF1 (3.10%)and on CONF2 data set (23.34%). Checking these presentations, the higher number of errors in CONF2 attributes to the use of many more videos and animations, some slides with very little content (including a couple of plain blue slides) and several cases of duplicated identical slides. For the last, we expect that by introducing temporal analysis into the matching process, taking into count the mostly sequential slides order during a typical presentation, , it will be possible to accurately label the identical slides.

We further tested the robustness of scene background matching to the choice of the matching threshold m. Figure 5 shows how the number of misses increases when the threshold is higher. The background matching with local keypoint

Frame	full-slide			small-slide			no-slide			Total		
Similarity	NCC1	NCC2	KP	NCC1	NCC2	KP	NCC1	NCC2	KP	NCC1	NCC2	KP
KP(100)	42/258	22/258	8/258	29/76	29/76	27/76	0/327	0/327	0/327	71/611	51/611	35/661
BP(100+400) Local	44/258	27/258	10/258	14/76	13/76	15/76	3/327	3/327	3/327	61/611	43/611	28/661

Table 3: Recognition error/total frames comparisons under two different similarity measures: NCC1 is without color correction, NCC2 is with color correction, and KP is Keypoint matching (no color correction is needed).

Dataset	Video	# full-slide	# small-slide	# no-slide	Total
CONF1	1	1/33	2/9	0/61	3/103
	2	2/76	3/3	0/72	5/151
	3	1/37	1/4	0/50	2/91
	4	0/18	4/6	0/23	4/47
	5	2/41	4/12	0/64	6/117
	6	2/53	13/42	0/57	15/152
	total	8/258 (3.10%)	27/76 (35.52%)	0/327 (0.00%)	35/661(5.29%)
CONF2	1	14/92	2/3	0/103	16/198
	2	9/58	1/1	0/104	10/163
	3	10/50	0/0	1/90	11/140
	4	27/40	1/1	0/61	28/102
	5	0/17	0/0	0/52	0/69
	total	60/257 (23.34%)	4/5 (80.00%)	1/410 (0.24%)	65/672(9.67%)
UNIV	1	0/48	9/39	0/66	9/153
	2	3/54	11/36	1/101	15/191
	total	3/102 (2.94%)	20/75 (26.66%)	1/167 (0.59%)	24/344(6.97%)

Table 4: Error rates of the baseline algorithm (i.e., no background matching), marked by the number of misrecongnized/total slides (error percentile) for full-slide , small-slide and no-slide frames, per each presentation. #iterations of the 1st RANSAC = 100.

Frame	full-slide			small-slide			no-slide			Total		
Alg.	CONF1	CONF2	UNIV	CONF1	CONF2	UNIV	CONF1	CONF2	UNIV	CONF1	CONF2	UNIV
KP(100)	8/258	60/257	3/102	27/76	4/5	20/75	0/327	1/410	1/167	35/661	65/672	24/344
KP(400)	9/258	56/257	3/102	24/76	4/5	17/75	0/327	1/410	1/167	33/661	61/672	21/344
BP(100+400) Global	8/258	60/257	3/102	25/76	4/5	16/75	0/327	1/410	1/167	33/661	65/672	20/344
BP(100+400) Local	10/258	**51/257**	3/102	**15/76**	4/5	**12/75**	3/327	**10/410**	1/167	28/661	65/672	16/344

Table 5: Performance comparison of four different algorithms; KP(100) and KP(400) denote the keypoint matching performance when the RANSAC is allowed for 100 and 400 iterations, respectively. The background matching, denoted as BP(), was tested twice, once in local-mode matching and second in global-mode matching. Each mode uses 100 RANSAC iterations in the first phase and 400 RANSAC iterations in the second phase, after background matching. The two modes are discussed in 2.1. Results on 3 different frame types are shown for the three data sets, CONF1, CONF2 and UNIV. For each method, frame type and data set, the number of errors/total frames of this class is displayed. The background matching with local search method outperforms the other methods.

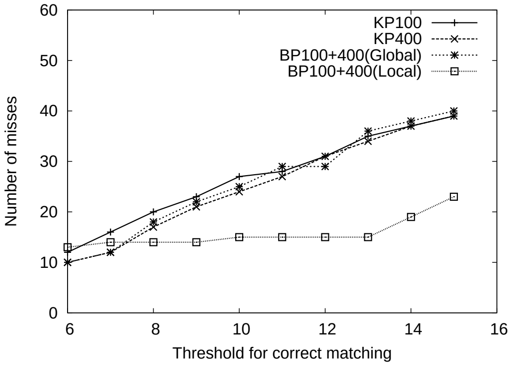


Figure 5: Algorithm robustness is demonstrated by repeating the experiment for different values of the matching acceptance threshold, m, and measuring the missrecognized small-slide frames. A higher threshold provides greater confidence in the matching. The background matching with local keypoint search, BP100+400(Local), outperforms the other three methods.

search algorithm outperforms all other three variants. It is nearly insensitive to the value of m, hence providing much better homographies and stronger matching for small slides than the alternatives. Changing m has no measurable impact on performance of full-slide matching (for which confidence levels are much higher), nor on no-slide frames classification (for which $m = 6$ already produces nearly perfect classification results, and increasing m only improves it).

6. CONCLUSIONS

We have demonstrated a comprehensive approach to matching slides to presentation videos. Experiments on three data sets show that the approach is viable for real world applications. We found that an implementation of keypoint matching together with constrained planar homography tuned to the application provided a very solid starting point for the development of our matching system. The unsupervised scene background matching is a robust and fast way to boost the performance of RANSAC in difficult cases such as small slides. We also show that searching initial keypoint matching locally can give more correct correspondences, thus making RANSAC more robust. Interestingly, the generally high performance of this initial version made measuring incremental improvement difficult in some cases.

In this work, we did not exploit the temporal information such as the usually sequential order of slide changes, which should be able to further help us resolve ambiguities between extremely similar or identical slides that often occur in a presentation. We intend to explore this direction in future work.

7. ACKNOWLEDGMENTS

We would like to thank many IBM colleagues who helped us collecting videos and slides data and generating manual labeling. We also extend our deep thanks to all the presenters who were willing to let us use their presentations for this study. This work was supported in part by the Arizona Center for Information Science and Technology (ACIST) and by IBM.

8. REFERENCES

[1] G. D. Abowd, C. G. Atkeson, A. Feinstein, C. E. Hmelo, R. Kooper, S. Long, N. N. Sawhney, and M. Tani. Teaching and learning as multimedia authoring: The classroom 2000 project. In *ACM Multimedia*, pages 187–198, 1996.

[2] A. Amir, G. Ashour, and S. Srinivasan. Automatic generation of conference video proceedings. In *Journal of Visual Communication and Image Representation, JVCI Special Issue on Multimedia Databases*, pages 467–488, 2004.

[3] A. Behera, D. Lalanne, and R. Ingold. Looking at projected documents: Event detection document identification., 2004.

[4] N. Christianini and J. Shawe. *Support Vector machines and other kernel-based learning method.* Cambridge University Press, 2002.

[5] B. Erol, J. J. Hull, and D. Lee. Linking multimedia presentations with their symbolic source documents: algorithm and applications. In *ACM Multimedia*, pages 498–507, 2003.

[6] S. Fathima, T. Mahmood. Indexing for topics in videos using foils. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages II: 312–319, 2000.

[7] A. Fischler, M. and C. Bolles, R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981.

[8] R. M. Haralick and G. S. Linda. *Computer and Robot Vision, Volume II.* Addison-Wesley, 1992.

[9] R. Hartley and A. Zisserman. *Multiple view and geometry in computer vision.* Cambridge University Press, 2002.

[10] T. Liu, R. Hjelsvold, and R. Kender, J. Analysis and enhancement of videos of electronic slide presentations. *IEEE International Conference on Multimedia and Expo (ICME)*, 2002.

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, pages 91–110, 2004.

[12] S. Mukhopadhyay and B. Smith. Passive capture and structuring of lectures. In *ACM Multimedia (1)*, pages 477–487, 1999.

[13] G. Pass, R. Zabih, and J. Miller. Comparing images using color coherence vectors. In *ACM Multimedia*, pages 65–73, 1996.

[14] L. A. Rowe and J. M. Gonzelez. Bmrc lecture browsers. In *http://bmrc.berkekey.edu/frame/projects/lb/index.html.*

[15] F. Wang, C.-W. Ngo, and T.-C. Pong. Synchronization of lecture videos and electronic slides by video text analysis. In *ACM Multimedia*, pages 315–318, 2003.