TEMPORAL MODELING OF SLIDE CHANGE IN PRESENTATION VIDEOS

Quanfu Fan (1), Arnon Amir (2), Kobus Barnard (1), Ranjini Swaminathan (1) and Alon Efrat (1)

(1) Department of Computer Science, University of Arizona, Tucson AZ85721
(2) IBM Almaden Research Center, 650 Harry Road, San Jose, CA95120

ABSTRACT

We develop a general framework to automatically match electronic slides to the videos of the corresponding presentations. The synchronized slides support indexing and browsing of educational and corporate digital video libraries. Our approach extends previous work that matches slides based on visual features alone, and integrates multiple cues to further improve performance in more difficult cases. We model slide change in a presentation with a dynamic Hidden Markov Model (HMM) that captures the temporal notion of slide change and whose transition probabilities are adapted locally by using the camera events in the inference process. Our results show that combining multiple cues in a state model can greatly improve the performance in ambiguous cases.

Index Terms— Algorithm, Cameras, Image matching

1. INTRODUCTION

Matching slides to videos provides an attractive way of indexing videos by slides for searching and browsing. It can also improve the quality of the videos through projecting the high-resolution slides back into the videos. Recently many approaches have been proposed to automatically match slides to videos [1, 2, 3, 4, 5, 6, 7, 8].

Depending on the capturing systems, the slides may appear dramatically differently in the video. A dynamic capturing system with one or more cameras that are allowed to pan, zoom and tilt has the flexibility to capture the presenter, the slides and the audience or all of them, thus produce more lively and instructional videos. However, the videos captured by such a system present various ambiguities between the captured slide images and the original slides, making the task of automating synchronization of slides with videos more difficult. Complications include zooming-in or zooming-out slides, slides partially occluded by the presenter, and no slide images due to the camera panning to the presenter or audience. In addition to these difficulties, some ambiguities come from the slides themselves such as identical slides and slide animation that sometimes generates extremely similar slides.

Although the dynamic nature of presentation video production poses additional challenges to slide-matching algorithms, it also yields useful cues on slide change, which can be used to help resolve ambiguities. For example, there usually is no slide change when the camera is zooming. Similarly, it is more likely that the camera will remain fixed when there is a slide change.

In this work, we extend our previous work [4] that matches slides based on visual features alone, and integrates the camera cue into a dynamic HMM in which the state transition probabilities are dependent on the camera events. The temporal model also captures the notion that slides are usually presented sequentially and not randomly, which as shown later in the paper, can greatly improve the ability of the model in disambiguating similar slides in the videos.

2. THE SLIDE-MATCHING FRAMEWORK

Our slide-matching framework consists of three phases: keypoint matching, camera event detection, and a dynamic HMM based on camera events (Fig. 1). In the first phase, frame-to-slide homographies (i.e the projection transformations between the slides captured by camera and their original ones), if available, are found by keypoint matching and all the frames are classified into 3 categories: *full-slide* (the entire frame shows the slide content), *small-slide* (the frame contains both a slide area and a substantial portion of the scene background), and *no-slide* (see [4] for more details). The second phase detects the camera event between each pair of consecutive frames by using the homographies computed and the frame types classified in the first phase. Finally, in the third phase, the visual features, temporal information and the camera events are incorporated into a dynamic HMM to find an optimal sequence of slides matching the frame sequence.

3. SLIDE EVENTS AND CAMERA EVENTS

In a presentation video there are events initiated by the presenter and the producer (camera person). These events determine the visual appearance of the video frames. For example, during a presentation, the presenter may make slide changes, write on the board, play video demonstrations, or browse the web. Accordingly, the producer may switch cameras, zoom or pan the camera in order to capture informative and meaningful scenes such as the slide content and the gestures of the presenter.

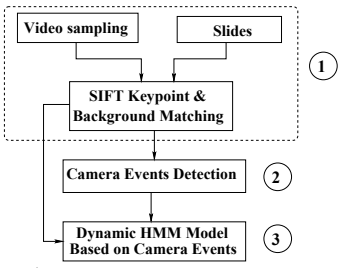

Fig. 1. The flow of our framework.

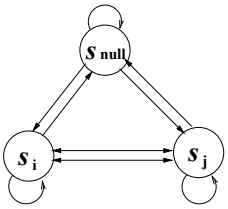

Fig. 2. A representation of state transition in our HMM. There is one *null* state between each pair of slides (s_i,s_j) to represent the no-slide frame.

Interestingly, these two types of events are not independent. For example, a camera zoom may indicate that the slide remains the same. Similarly, the camera is less likely to change during a video demo or slide change. More generally, we expect the frame-to-slide homography to remain across slide changes and the slide to remain unchanged across camera changes.

An event from the presenter is called a *slide event*, which describes how slides change in an presentation. We denote by $\mathbf{S_k}$ a slide change event, when the current slide, s_i, is immediately followed up by slide s_{i+k} (a k slides "jump"). The sign of k indicates the direction of the slide change and $k = 0$ implies that the current slide stays unchanged. To model events involving frames with no slides, we use $\mathbf{S_{span}}$ to denote a change from seeing a slide to seeing no slide, and $\mathbf{S_{noslide}}$ to denote the case of a two no-slide continuum.

A camera event describes how the producer operates the cameras when capturing a presentation. Basic camera operations include zooming, staying fixed and panning/tilting. In this paper, we define 6 types of camera operations of interest: *zoom-in, zoom-out, stay-fixed, slide-in, slide-out* and *stay-out*. Note that some of our definitions here are slightly different from what a reader may know already. *Zoom-in* magnifies the slide area significantly in the current frame with respect to the previous frame. It happens when the producer increases the focal length or switches to a camera with a longer focal length. *Zoom-out* is defined inversely. *Stay-fixed* refers to a static status of the camera when it focuses on the slides (either small or full slides) without movement. The other 3 events relate to camera panning or tilting. When the camera moves from the slide to capture the presenter/audience only (no slide in the frame), we call it *slide-out*. The opposite operation is defined as *slide-in*. *Stay-out* is the camera event between *slide-out* and *slide-in* when the slide is not being captured. A camera in *stay-out* may still zoom or move. We do not further differentiate between them as they provide little information about slide change.

4. DETECTING CAMERA EVENTS

Camera motion can be considered as an optical flow problem. Many approaches have been developed to detect camera operations based on the analysis of the motion vector field (see [9] for a review). In our case, the previously computed frame-to-slide homographies can be used to spot the slide position in the frames. This leads us to simply represent each of the three events (*zoom-in*, *zoom-out* and *stay-zoom*) that ensure the existence of frame-to-slide homographies by a Gaussian distribution over the ratio of the slide areas of two consecutive frames. The parameters of the distributions were fitted directly from a held-out portion of the ground truth data. This method yielded over 99% classification accuracy. In the case where no homography is available, we use the frame types classified by the matching algorithm to determine the camera events. For example, a current small-slide frame and a following no-slide frame indicate a slide-out event between the two frames.

5. TEMPORAL MODELING OF SLIDE CHANGE

We first describe a standard HMM without using camera information to model slide change. We then extend it to a dynamic HMM in which the model parameters are adjusted locally by using the camera events.

5.1. A Standard HMM Without Camera Information

Slides do not change randomly in a presentation. Instead, they almost always advance sequentially according to their order in the presentation file, though sometimes the sequence may be interrupted by shifting to the previous slide or jumping to an arbitrary slide. To capture this notion, we model slide change by a HMM with slide numbers as hidden states. Since the number of slides for each presentation can vary greatly, we consider the slide transition as *stateless*, i.e we assume that the transition from slide 2 to slide 3, for instance, is no different from a change from slide 7 to 8. We also introduce an auxiliary state *"null"* (Fig. 2) between each pair of slides (s_i, s_j) to represent the no-slide frame. Note that the overhead for adding a *"null"* node between each pair of slides is negligible as only one *"null"* node needs to be actually maintained in the implementation due to the stateless assumption of the slide transition. We estimate the stateless slide transition probabilities from held out data. Because the data is

limited, we enforce smoothness using a Poisson distribution as follows,

$$A(s_i|s_j) = \begin{cases} \eta P(-m,\lambda) & m<0 \\ P(0,\lambda) & m=0 \\ (1-\eta)P(m,\lambda) & m>0 \end{cases} \quad (1)$$

where $m = s_i - s_j$ is the slide event, and $P(x,\lambda) = \frac{e^{-\lambda}\lambda^x}{x!}$ is the Poisson distribution. η is the frequency of slides going backwards with respect to going forwards. Both η and λ were fitted from a held-out portion of the ground truth data. Values for η were between 0.05 and 0.09 and values for λ were between 0.04 and 0.05. Table 4 shows the actual distribution of slide transition in our data. Because of the high frame sampling rate we use (1 frame/ sec), there is a very high chance that a slide stays unchanged. It also appears, as expected, that slides tend to change forward much more frequently than backward.

We estimate the slide observation probability $B(f_i|s_j)$ of a frame f_i given a slide s_j by the fraction of the matched keypoints of f_i to s_j over the total number of matched keypoints of f_i to all the slides. When no matching slides are found for a frame that is marked as a slide frame by the slide matching algorithm, a uniform probability is assigned.

The optimal sequence of slides matching the frame sequence is found by the well-known Viterbi algorithm [10].

5.2. A Dynamic HMM Based on Camera Events

In the standard HMM, the model parameters are derived in the training data and the parameters remain unchanged during the entire inference process. The dynamic model can be regarded as incorporating context dependent information into the transition probabilities [11]. In our case, the context dependent information is the relationship between slide events and camera events. For example, camera change is associated with a higher probability that there is no slide change.

More specifically, we condition the state transition probability through the camera-dependent transition probability $p(s_i|s_j,c)$. These probabilities are estimated from the held-out portion of the ground truth data. A trivial modification of the Viterbi algorithm, namely replacing $p(s_i|s_j)$ by $p(s_i|s_j,c)$ at each time step based on the camera event c, is sufficient to find an optimal slide sequence. Nevertheless, because a specific camera event may exclude many states that have to be visited in the standard HMM, we can employ a variant of Viterbi algorithm such as the one used in [12] to speed up the inference. For example, if we know a camera is at the *stay-out* state (not looking at the slide), the algorithm only needs to visit the *"null"* node instead of all nodes.

6. EXPERIMENTS AND RESULTS

The videos used for our evaluation are $CONF1$ and $UNIV$ (see [4] for more details). $CONF1$ is a set of 6 videos captured by three cameras from a corporate conference. $UNIV$ has 3 videos captured by two cameras in a university seminar. We first sample the videos by extracting one frame per second. Additional keyframes are extracted as determined by shot boundary detection algorithm, to avoid missing consecutive fast changes. We then manually construct a ground truth matching between frames and slides and the camera events.

We use two evaluation methods in our experiments. First, as in previous work [4], we consider the number of misrecognized frames over the total number of frames. However, due to the higher sampling rate used here, the error computation is biased towards slides that appeared for a longer time. Thus, we use a second method that evaluate the algorithms based on video segments. A segment is defined as a video clip with neither slide change nor camera change. The error rate for a segment is defined as,

$$\mathbf{e} = \frac{\text{\# of incorrectly identified frames in the segment}}{\text{\# of total frames in the segment}} \quad (2)$$

Segments with less than 2 frames (2 seconds) were ignored in the experiments.

We measure the performance of three algorithms: the keypoint matching algorithm (BASE), the standard HMM (HMM) and the camera-event-based HMM (CHMM).

The results on the two data sets are presented in Table 1 and 2. Both HMMs greatly outperform the base matching algorithm and CHMM performs the best, showing clearly the advantage of using the temporal and camera information. As we expected, there was significant improvement in the matching performance of small slides. In addition, the matching performance for large slides also improved.

On these two data sets CHMM performs slightly better than HMM (comparable on UNIV and some improvement on CONF1). The results are consistent with the observation that the density and complexity of the camera events in UNIV is relatively low, and even in CONF1 they are far from extreme. We thus expect more improvement on more difficult data.

We conduct another experiment on CONF1 to see how much the temporal and camera cues contribute to the performance improvement in the case of small slides. To do this we ignore the key point matching cues and compute the alignment based only on the temporal and camera event model. As recorded in Table 3, there is still more than 60% accuracy on the small slides for both models even if no slide keypoint matching information is used. This further demonstrates the potential contribution of temporal models to a robust slide-matching system.

Finally, we broke down the results in Table 5 according to the slide events. The results clearly show that the HMMs can model the sequential change of slide very well. The HMMs also showed the potential to handle non-sequential slide change on CONF1, but failed on KUAT due to very limited examples of no-sequential change in the data.

Data	Alg	# full-slide	# small-slide	# no-slide	Total
CONF1	BASE	132 (2.15)	193 (12.53)	**7** (0.09)	332 (2.15)
	HMM	104 (1.69)	88 (5.71)	9 (0.12)	201 (1.30)
	CHMM	**97** (1.58)	**84** (5.46)	11 (0.14)	**192** (1.24)
	# frames	6147	1540	7742	15429
UNIV	BASE	85 (2.37)	136 (23.09)	**70** (1.83)	291 (3.64)
	HMM	28 (0.78)	42 (7.13)	116 (3.03)	**186** (2.32)
	CHMM	28 (0.78)	**40** (6.79)	120 (3.133)	188 (2.35)
	# frames	3586	589	3830	8005

Table 1. **Frame-oriented** overall error rates of the three algorithms, marked by the number of mis-recognized frames and the error percentage in the brackets on the full-slide , small-slide ,and no-slide frames.

Data	Alg.	# full-slide	# small-slide	# no-slide	Total
CONF1	BASE	7.95 (2.89)	13.41 (15.96)	0.06 (0.02)	21.42 (3.50)
	HMM	8.46 (3.08)	5.04 (6.00)	0.29 (0.11)	13.79 (2.25)
	CHMM	**7.71** (2.80)	**4.84** (5.76)	0.29 (0.12)	**12.85** (2.10)
	# segments	275	84	253	612
UNIV	BASE	3.15 (2.05)	14.93 (19.14)	**1.23** (0.99)	19.31 (5.41)
	HMM	0.38 (0.25)	3.71 (4.76)	1.45 (1.16)	5.54 (1.55)
	CHMM	0.38 (0.25)	**3.53** (4.52)	1.51 (1.21)	**5.42** (1.52)
	# segments	154	78	125	357

Table 2. **Segment-oriented** overall error rates of the three algorithms, marked by the number of mis-recognized segments and the error percentage in the brackets on the full-slide , small-slide ,and no-slide frames.

Alg.	RAND	HMM	DHMM
Err.	81.9/84 (98.0)	30.0/84 (35.7)	27.9/84 (33.2)

Table 3. **Segment-oriented** overall error rates of small slides of the 2 HMM models on CONF1, computed by assigning uniform observation probability to all small slides classified from the keypoint matching algorithm. The numbers in the brackets are error percentage. The very high error rate in the first column reflects a random guess of the slide numbers.

Data	$\mathbf{S_{k<-1}}$	$\mathbf{S_{-1}}$	$\mathbf{S_0}$	$\mathbf{S_1}$	$\mathbf{S_{k>1}}$	$\mathbf{S_{span}}$	$\mathbf{S_{noslide}}$
CONF1	2	5	7477	181	17	251	7484
	(0.01)	(0.03)	(48.50)	(1.17)	(0.11)	(1.63)	(48.54)
KUAT	1	8	4026	139	1	127	3700
	(0.01)	(0.10)	(50.31)	(1.74)	(0.01)	(1.59)	(46.24)

Table 4. The distribution (%) of different slide-change events in our data. The high probability of a slide staying unchanged is caused by the high frame sampling rate used in our experiments. It also appears that slides tend to go forward much more frequently than backward, as anticipated.

Data	$\mathbf{S_{k<-1}}$	$\mathbf{S_{-1}}$	$\mathbf{S_0}$	$\mathbf{S_1}$	$\mathbf{S_{k>1}}$	$\mathbf{S_{span}}$	$\mathbf{S_{noslide}}$
CONF1	100	20	2.38	9.39	17.65	5.18	0.21
KUAT	100	100	1.56	9.35	100	8.66	3.51

Table 5. The error percentage of the mis-recognized slide events in the case of CHMM .

7. CONCLUSIONS

We present a general framework to automatically match slides to presentation videos with high accuracy. Our results suggest that both the temporal and camera cues are very promising sources of information to disambiguate the occurrence and identity of slides in videos when conditions are challenging. Further complexity in the inter-relation of these events could be modeled using coupled HMM's [13]. We are currently exploiting this approach.

8. REFERENCES

[1] A Behera, D Lalanne, and R Ingold, "Looking at projected documents: Event detection document identification," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2004.

[2] Y. Chen and W. J. Heng, "Automatic synchronization of speech transcript and slides in presentation," in *International Symposium on Circuits and Systems (ISCAS)*, 2003, pp. 568–571.

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

[4] Q. Fan, K. Barnard, A. Amir, A. Efrat, and M. Lin, "Matching slides to presentation videos using sift and scene background matching," in *the 8th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR)*, 2006.

[5] T. F. Syeda-Mahmood, "Indexing for topics in videos using foils," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2000, pp. II: 312–319.

[6] 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.

[7] S. Mukhopadhyay and B. Smith, "Passive capture and structuring of lectures," in *ACM Multimedia (1)*, 1999, pp. 477–487.

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

[9] J. Korpi-Anttila, "Automatic color enhancement and scence change detection of digital videos," *Licentiate thesis, Helsinki University of Technology*, pp. 37–40, 2003.

[10] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, pp. 260–267, 1967.

[11] S. Young, J. Odell, and P. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings ARPA Workshop on Human Language Technology*, 1994, pp. 307–312.

[12] J. Kohlmorgen and S. Lemm, "A dynamic hmm for on-line segmentation of sequential data," in *NIPS*, 2001.

[13] M. Brand, N. Oliver, and A. Pentland, "Coupled hmm for complex action recognition," in *Proc. of Conference on Computer Vision and Pattern Recognition*, 1997, vol. 29, pp. 213–244.