A ROBUST FRAMEWORK FOR ALIGNING LECTURE SLIDES WITH VIDEO

Wang Xiangyu, Subramanian Ramanathan, Mohan Kankanhalli

School of Computing, National University of Singapore.

ABSTRACT

We propose a robust approach for aligning lecture slides with lecture videos using a combination of Hough transform, optical flow and Gabor analysis. A Markov Decision Process model is used to incorporate prior knowledge for enhanced recognition. We demonstrate synchronization of slides with videos containing de-focused slide content, speaker occlusion as well as camera pan, tilt and zoom sequences. Experimental results confirm the effectiveness of our approach for multimedia indexing applications.

Index Terms— Robust synchronization, Multimedia Indexing, Hough transform, Optical-flow, Gabor analysis, Markov Decision Process

1. INTRODUCTION

The advent of digital libraries to manage multimedia collections has greatly increased the need for multimedia indexing, which is the key to content-based retrieval and distance learning. A basic requirement of digital libraries is the ability to associate event-specific data, an example of which involves *linking of topics/contents in lecture slides to corresponding segments of the lecture video*. Video-slide synchronization facilitates instant referencing, as the user can directly access the video segment of interest and conversely, availability of the relevant slide content along with the video enhances the user understanding as well as the overall experience.

Substantial research has focused on video-slide alignment in the past. Pioneering works such as Classroom 2000 (C2K) [1] manually align slides with video segments using timestamps, which is time and labor-intensive. Synchronization performed on the basis of audio cues in [2] aids generation of new slides to support impromptu speech, but requires manual processing for proper alignment. More reliable visual cues are employed for automatically aligning slides with video in [3, 4]. In [3], the video is divided into homogeneous slide segments using a feature-based algorithm, followed by slide matching based on a Hausdorff distance-based similarity metric. Detection of slide regions in video is accomplished using illumination-invariant background color description in [4], and region hashing is employed to recognize the video slide content.

However, most of these these algorithms discuss slide alignment with passively captured video where the slide content is clearly visible. Recently, [5] discusses a dynamic Hidden Markov Model (HMM) framework for matching slides with videos containing camera events (pan, tilt and zoom sequences). This paper proposes a robust framework for video-slide synchronization to handle de-focused slide content, speaker occlusion and camera events, examples of which are shown in Fig. 1. The tackled cases in different methods is shown in table 1. Our proposed method handles all the mentioned situations while the previous algorithms only handle parts of them.

(a) (b)

(c) (d) (e)

Fig. 1. (a) Video #1 contains clear and easily extractable, but animated slide content. (b), (c) and (d) illustrate more difficult cases. (b) Video #2- De-focused Slide content captured by static camera. (c) Video #3 - Speaker occlusion of perspectively rotated projector screen. (d),(e) are frames from a *zoom-in* sequence in Video #4.

Methods	camera event	speaker occlusion	de-focusing
[4]	No	No	Yes
[5]	Yes	Yes	No
Proposed	Yes	Yes	Yes

Table 1. Comparison of the methods

2. ALGORITHM DESCRIPTION

Our proposed video-slide synchronization algorithm is outlined in Fig. 2. For every frame $f_i, i = 1..N$, in the presentation video consisting of N frames, the algorithm computes position of captured slide in the frame and its best match

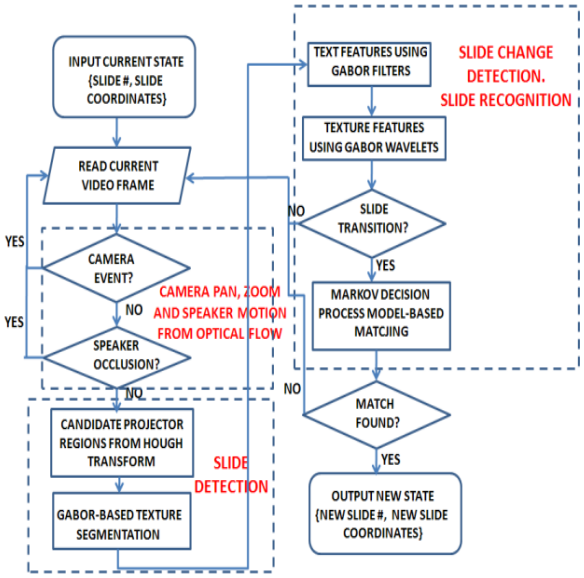


Fig. 2. Overview of the proposed framework

counterpart, S_j, in a sequence of M slides, $S_1, .., S_M$. Given the current state defined by frame slide coordinates and slide index (denoted by #), we first use optical flow cues to detect camera events or slide occlusion in video to ensure proper alignment. This is because slide transitions are unlikely during camera events [5] while extraction and recognition of the frame slide region is difficult in the presence of occlusion. In the absence of both, we extract the frame slide region using a two-step procedure.

Evidently, every video frame can be classified as either *small-slide* (captured slide occupies part of the frame), *full-slide* (slide completely occupies the frame) or *no-slide* (frame contains no slide content). As a pre-processing step, we apply Hough transform to extract quadrilaterals in *small-slide* frames as candidate slide regions. Hough extraction helps avoid expensive computations for frame slide region detection, and if no candidates are found, reduces subsequent frame classification to only *full-slide* or *no-slide*. In the second step, we precisely identify the slide region from among the candidates and determine its coordinates, or classify the frame as *full-slide* or *no-slide* using Gabor texture analysis.

Text as well as texture features are employed to match the detected frame slide to its counterpart for recognition. Gabor filter banks are useful for extracting high frequency slide text regions [6] while Gabor texture features can achieve rotation-invariant texture matching [7]. Also, incorporating prior knowledge that slides usually advance sequentially during the presentation can improve recognition rates, especially when the presentation contains similar-looking or animated slides. We employ prior knowledge using a Markov Decision Process model to guide recognition. Our experimental results confirm that reliable video-slide synchronization is possible for most challenging cases using the proposed approach. The paper is organized as follows. The next section elaborates the various steps involved in slide detection and recognition, while results and discussion are presented in Section 3. Section 4 outlines conclusions and future work.

Given the current frame slide position and slide index (both of which are NULL at the beginning of the presentation), we first check for camera events (pan, tilt, zoom) between the previous and current video frames via optical flow. We use the motion cooccurrence-based homogeneous motion detection algorithm [8] for identifying possible camera events. Since the likelihood of a slide transition (and thereby, a change in video-slide alignment) during camera events is minimal, we ignore camera event sequences as possible candidates where a new slide may be presented.

Next, we look for possible slide occlusion due to speaker motion. In the absence of any camera event, optical flow mainly corresponds to motion of the presenter, since foreground changes owing to speaker motion are more pronounced than slide transition-related background changes. Any speaker motion within the frame slide region corresponds to slide occlusion by the presenter. Since detection and identification of the slide content under occlusion is difficult, and the co-occurrence of slide occlusion and slide transition is generally implausible, slide-occluded frames are also discarded from further processing. Therefore, we will attempt to detect and recognize slide content for synchronization only in the absence of a camera event or upon de-occlusion of the frame slide region.

2.1. Slide detection

In order to localize probable frame slide regions for further processing, we perform Hough extraction of quadrilaterals in the video frame to identify candidate slide regions. Detection of vertical image edges followed by near-horizontal edges in their vicinity helps isolate slide candidates in a robust manner. Hough extraction can successfully isolate slide candidates in *small-slide* frames as shown in Fig. 3 (a),(g). While our algorithm doesn't assume a bounded frame slide region, Hough extraction as a pre-processing step to slide detection has two advantages - (i) it eliminates the need to perform expensive computations as in [4], since only the candidate image regions need to be processed further and (ii) a binary classifier is sufficient to detect *full-slide* or *no-slide* frames if no candidates are identified. Gabor texture analysis is then employed to precisely isolate the most probable frame slide region from among the candidates or determine *full-slide* and *no-slide* frames.

2.1.1. Gabor filters and wavelets

A 2D Gabor filter $G(x, y)$ is a Gaussian modulated by a sinusoid.

$$G(x,y) = g_\sigma(x,y)exp(2\pi jU(xcos\theta + ysin\theta)), \text{where} \quad (1)$$

$$g_\sigma(x,y) = \frac{1}{2\pi\sigma_x\sigma_y}exp(\frac{-1}{2}(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2})) \quad (2)$$

Gabor filter parameters are the radial frequency U, the orientation θ and the Gaussian width σ. A Gabor filter bank is a collection of Gabor filters with different U's and θ's, while Gabor wavelets are generated from the dilation and rotation of the mother wavelet $\psi = G(x, y)$, through the generating function

$$\psi_{pq}(x, y) = a^{-p}\psi(\bar{x}, \bar{y}), \text{where } p = 0..P-1, q = 0..Q-1 \quad (3)$$

are the scale and orientation factors respectively, with P and Q respectively being the total number of scales and orientations. Here,

$$(\bar{x}, \bar{y})^T = R(x, y)^T \quad (4)$$

where $(.)^T$ denotes transpose, $R = \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix}$, $a > 1$ and $\theta = \frac{q\pi}{Q}$. Convolving an image with $G(x, y)$ yields the Gabor filtered output while the discrete Gabor wavelet transform of an image, $G_{p,q}(x, y)$, is given by its convolution with $\psi_{pq}(x, y)$.

2.1.2. Slide texture segmentation

Upon convolving the candidate slide regions (or the complete video frame) of dimensions $M \times N$ with the Gabor filters at multiple scales and orientations, the image energy at scale p and orientation q is given by $E(p, q) = \sum_x \sum_y |G_{pq}(x, y)|$. The mean, $\mu_{p,q}$, and standard deviation, $\sigma_{p,q}$, that represent the image texture features at p, q are given by

$$\mu_{pq} = \frac{E(p, q)}{MN}, \ \sigma_{pq} = \sqrt{\frac{\sum\sum |G_{pq}(x, y) - \mu(p, q)|}{MN}} \quad (5)$$

while the average image energy μ over all scales and orientations is given by

$$\mu = \frac{\sum_{p,q}(\mu_{pq})}{pq} \quad (6)$$

We find that μ, with $p = 4$ and $q = 3$, can be used to reliably identify uniformly textured frame slide regions. k-means clustering of μ values can reliably classify *slide* frames from *no-slide* frames (Fig. 3(d)). We employ the μ feature to detect the most probable slide region from among the candidates as well as distinguish between *full-slide* and *no-slide* frames (Fig. 3 (b),(c),(f),(g)). Upon determining the most probable slide region, we use Gabor filter-based texture segmentation [9] to precisely determine frame slide coordinates followed by perspective correction (Fig. 3(h)), to obtain the frame slide region for matching.

2.2. Slide change detection and recognition

A rotation-invariant texture classification scheme using (μ_{pq}, σ_{pq}) features is proposed in [7]. However, due to significant differences in the appearance of extracted frame slide regions and slide images (frame slide regions are usually blurred) and the presence of similar-looking slides, directly matching the frame slide images with the original slides, based on minimum Euclidian distance, is inefficient. Therefore, we reduce the number of candidate matches for the extracted frame slide using text features while including those slides predicted as probable matches by the Markov Decision Process (MDP) model described below.

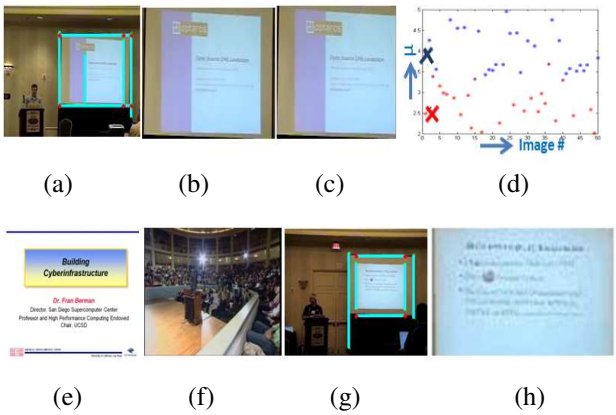


Fig. 3. (a) Hough transform extracted slide coordinates (in red)- green lines denote detected vertical and horizontal edges.(b) Region #1-$\mu = 3.7$ and (c) Region #2- $\mu = 3.85$. (d) k-means clustering ($k = 2$) of μ values for 50 *slide* (blue) and *no-slide* (red) images. Red and blue crosses denote centroidal values. (e,f) $\mu = 5, 2$ for *full-slide*, *no-slide* frames. Gabor texture segmentation on (g) Hough transform result yields (h) Slide region in video.

Extraction of high frequency text regions using Gabor filter banks is proposed in [6] and is found to be robust for high resolution image and video data. Results of text extraction for a *text-only*, *text+image* and *image-only* slide using the above procedure is shown in Fig. 4. We retrieve the top five slide image matches for the extracted frame slide region based on their closeness to the text feature vector $t_f = \{y_l, n_l, m_l\}$, where y_l, n_l and m_l denote the y position of the first sentence, number of sentences and maximum sentence length respectively. Even though features extracted from low-resolution frame slides are relatively noisy, text analysis helps exclude slide images that are vastly different from the frame slide, from the matching process. Also, content change in the extracted frame slide is assumed whenever there is significant error between consecutive text-extracted binary images or a considerable change in the Gabor texture feature vector over time.

2.2.1. Markov Decision Process model

Markov decision processes (MDPs) are used for optimal decision making in an accessible, stochastic environment. Given the 4-tuple, (S, A, M_{ij}^a, R_i^a), where S denotes the state space, A denotes possible set of actions in each state, M_{ij}^a denotes probability of transiting to state j upon undertaking a in state i and R_i^a denotes the reward for undertaking a in state i, the MDP model computes the sequence of optimal actions at every state, known as *policy*. For the slide matching problem, the number of states is equal to the slide count, and as the MDP model is invoked upon detecting a slide change, possible actions at every state include transiting to the next slide/the previous slide/an arbitrary slide in the presentation

sequence. As verified in [5], slides generally advance sequentially, *i.e.*, upon presentation of the i^{th} slide S_i, S_{i+1} and S_{i-1} are most likely to be shown next in the decreasing order of probability, with the chance of transiting to an arbitrary slide being minimal.

If $A = \{T_k\}$, where T_k denotes transition by k slides from S_i, in a presentation sequence of N slides, prior knowledge is incorporated into the MDP model by setting M_{ij}^a's to 1 (deterministic) and the reward function

$$R_i^{T_k} = \begin{cases} (1-\eta)P(k,\lambda), & k \geq 1 \\ \eta P(-k,\lambda), & k \leq -1 \end{cases} \quad (7)$$

with $\eta = 0.9, \lambda = 0.1$ and $P(x,\lambda) = \frac{e^{-x}(\lambda^x)}{x!}$, the Poisson distribution. When the text features are noisy, the most probable matches determined by the MDP model are included among the candidates for texture recognition.

3. RESULTS AND DISCUSSION

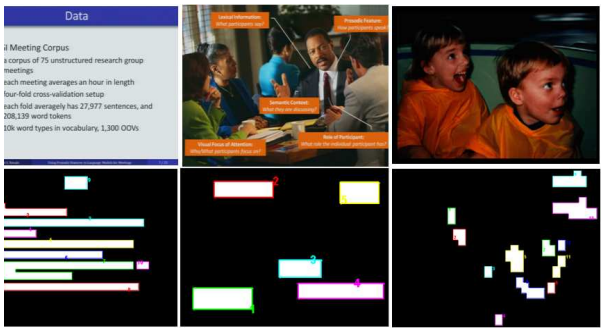
Video	#Slides	#Frames	Transitions		Matches	
			Act	Det	Cor	Wrong
#1	15	800	17	25	20	5
#2	14	5000	13	8	4	4
#3	10	2000	9	12	11	1
#4	38	2700	39	59	45	14

Table 2. Synchronization performance on test sequences

Video #1 contains animations, Video #2 contains de-focused slides, Video #3 contains speaker occlusion as well as camera events (pan and zoom), while the camera typically switches between speaker (*no-slide*) and the presentation slide (*full-slide*) with occasional *zoom-in* and *zoom-out* for Video #4, which also contains animations. Table 2 presents experimental results obtained using the proposed video-slide synchronization for the test sequences illustrated in Fig. 1. Table columns 4-7 compare our algorithm performance against ground truth. While columns 4 and 5 compare the actual and detected number of slide transitions, columns 6-7 present the number of correct and wrong video-slide matches upon slide change detection. Even though some spurious slide transitions are detected, all original frame slide changes are correctly identified for videos #1, #3 and #4 using the proposed approach. Generally over 75% matching accuracy is achieved. Matching errors for Video #1 are primarily observed for animated content, while for Video #2, texture recognition is unreliable for de-focused *text* content and works only for *image/figure* slides. Mismatches are again observed for animations in Video #4, while for Video #3 which contains many *image* slides, recognition accuracy is high (91.7%).

4. CONCLUSION AND FUTURE WORK

The proposed framework has been found to reliably segment, classify and recognize frame slide regions in challenging presentation videos. While experimental results confirm that slide transitions in video can be correctly identified, matching is affected by slide animations and de-focusing. Future work involves OCR implementation along with use of features (like SIFT) to achieve robust slide matching.



(a) (b) (c)

Fig. 4. Original slides (top-row) and extracted text regions using Gabor filter bank (bottom row) for a (a) *text-only* (b) *text+image* and (c) *image-only* slide. (bottom row).

5. REFERENCES

[1] G. Abowd, C. Atkenson, A. Feinstein, C. Hmelo, R. Kooper, S. Long, N. Sawhney, and M. Tani, "Teaching and learning as multimedia authoring: The classroom 2000 project," in *ACM-MM*, 1996, pp. 187–198.

[2] Gareth J. F. Jones and Richard J. Edens, "Automated alignment and annotation of audio-visual presentations," *Lecture Notes in Computer Science*, vol. 2458, pp. 187–196, 2002.

[3] S. Mukhopadhyay and Brian C. Smith, "Passive capture and structuring of lectures," in *ACM MM*, 1999, pp. 477–487.

[4] T. F. Syeda Mahmood, "Indexing for topics in videos using foils," in *CVPR*, 2000, pp. 312–319.

[5] Q. Fan, A. Amir, K. Barnard, R. Swaminathan, and A. Efrat, "Temporal modeling of slide change in presentation videos," in *ICASSP*, April 2007, vol. 1, pp. I–989–I–992.

[6] S.S. Raju, P.B. Pati, and A.G. Ramakrishnan, "Text localization and extraction from complex color images," 2005, pp. 486–493.

[7] S. Arivazhagan, L. Ganesan, and S. Padam Priyal, "Texture classification using gabor wavelets based rotation invariant features," *PR Letters*, vol. 27, no. 16, pp. 1976–1982, 2006.

[8] Hyun-Ho Jeon, A. Basso, and Peter F. Driessen, "Camera motion detection in video sequences using motion cooccurrences," in *PCM (1)*, 2005, vol. 3767, pp. 524–534.

[9] Anil K. Jain and Farshid Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pat. Rec.*, vol. 24, no. 12, pp. 1167–1186, 1991.