Sparse Time-Varying Graphs for Slide Transition Detection in Lecture Videos

Zhijin Liu[1,2], Kai Li[1,2,*], Liquan Shen[1,2], and Ping An[1,2]

[1] School of Communication and Information Engineering, Shanghai University, Shanghai, China
[2] Key Laboratory of Advanced Displays and System Application, Ministry of Education, Shanghai, China
* Corresponding author (email: kailee@shu.edu.cn)

Abstract. In this paper, we present an approach for detecting slide transitions in lectures videos by introducing sparse time-varying graphs. Given a lecture video which records the digital slides, the speaker, and the audience by multiple cameras, our goal is to find the keyframes where slide content changes. Specifically, we first partition the lecture video into short segments through feature detection and matching. By constructing a sparse graph at each moment with short video segments as nodes, we formulate the detection problem as a graph inference issue. A set of adjacency matrix between edges, which are sparse and time-varying, are then solved through a global optimization algorithm. Consequently, the changes between adjacency matrix reflect the slide transition. Experimental results show that the proposed system achieves the better accuracy than other video summarization and slide progression detection approaches.

Keywords: Lecture video, slide transition, sparse time-varying graph.

1 Introduction

Nowadays, e-learning has become an important learning means to acquire knowledge. A large number of lecture videos are posted on the Internet everyday, and most of these videos are unstructured. If users want to find some specific knowledge, they usually have to browse the entire video, which is time-consuming. Therefore, it is essential to automatically extract the representative summary for lecture videos.

Detection of slides transition is a critical issue for lecture video summarization. Various lecture videos can capture the projected slides and the speaker by a pan-tilt-zoom (PTZ) camera, record the computer screen directly, or even switch from the PTZ camera to the screen recorder. During this process, the slide content may remain the same for a long time or change to others quickly, while the users have to watch the whole video to have these findings. Apparently, the appearance change in the video frame does not necessarily indicate a slide transition. It is hard to tell the real slide transition from the disturbance like camera motion, camera switch, and people movement.

Unfortunately, previous approaches usually extracted slide transition keyframes by measuring the visual difference between adjacent frames. Various features such as histogram [14], SIFT [2], and wavelet [19] are chosen to describe the appearance similarity. These methods fail to deal with frames which contain people movement and camera motion, which are common noise interruption in some types of lecture video. Recently, an iterative approach has been proposed to localize the projection screen and detect the slide transition [9]. However, it is targeted for a specific type of lecture video which capture the projection screen by a single PTZ camera. Camera switches are not allowed, thus its application range is limited.

In this paper, we present an automatic approach to detect slide transitions in lecture videos by inferring sparse time-varying graphs. We first partition the video into several small segments by feature detection and matching. Inspired by the storyline summarization approach [8], we regard each segment as a node to construct a sparse time-varying graph. This graph is able to model the transition from one segment (slide) to another. After inferring the adjacency matrix of the graph through an global optimization, we analyze them to generate the slide transition keyframes robustly.

For evaluation, we collect a variety of lecture videos and compare our system with general video summarization approaches and slide progression detection method. Experimental results show that our system is able to handle several types of lecture videos and achieve the best performance.

The remainder of this paper is organized as follows. Sec. 2 investigates the lecture video summarization problem. Sec. 3 describes the system overview and general pipeline of each part. Sec. 4 presents the core sparse time-varying graph. Sec. 5 shows the experimental results and verifies the superiority of our system. Sec. 6 concludes this paper.

2 Related Work

In this section, we mainly review the literatures in lecture video summarization.

Many previous approaches generate the summary by measuring the appearance similarity between two adjacent frames. To compute the similarity, several features are extracted, such as color histograms, corner points, and edge information. For instance, some algorithms [14], [10] leveraged color histogram to summarize lecture videos. Jeong et al. [6] detected the forward and backward slides change by a recursive pruning algorithm. However, on one hand, if there are camera motions in the lecture video, appearance difference is not always effective; on the other hand, parameter estimation for the similarity threshold is trivial.

To address these problems, some shot boundary detection approaches were proposed. For example, Zhao et.al [22] presented a shot boundary detection algorithm based on fuzzy theory. They segmented videos into six different classes to detect shot boundary, and trained the camera movement features to avoid interference. Porter [17] designed a shot segmentation algorithm by a two-component frame differencing metric. Other classification methods introduced Support Vector Machines [21] and Neural Networks [13] to recognize shot boundaries. Recently, Li et al. [9] tracked the feature trajectories to find the slide progression frames.

Some work employed additional data to obtain a video summary, such as text embedded in lecture videos [20], [16], audio signal [5],[18], and electronic slides [1]. Wang et al. [20] reconstructed high-resolution video texts to detect and analyze text information for matching video clips. Ngo et al. [16] employed a foreground vs. background segmentation algorithm to obtain the projected electronic slides, and then detected and analyzed text captions to detect slide transitions. In their experiments, the camera is fixed and remain stationary, which is not universal applicable. Fan et al. [1] matched original electronic slides to presentation videos by a hidden Markov model. Since their input is the lecture video with its original electronic slides, which is not suitable for mostly lecture video summarization. In contrast, our method automatically detects slide transitions without additional data.

Our work is inspired by but different from the sparse time-varying graphs for reconstructing storyline graphs of videos and photos [7], [8]. [8] used a set of photo streams to construct a storyline summary that represents the narrative structure of activities by inferring sparse time-varying graphs. The storyline can be further used for sequential image prediction. [7] proposed a scalable approach to jointly summarize a set of associated videos and images. Temporal graph analysis was also considered in [15]. The difference is that the temporal graph is used for scene modeling and detection. In our system, we introduce the temporal graphs to infer the real slide changes, while the graph in [8] is targeted to discover the common structure of an event taken by many people.

3 Problem Settings

The input to our system is a lecture video which is represented by a set of frames $\mathcal{I} = \{\boldsymbol{I}^1, \cdots, \boldsymbol{I}^N\}$, where N is the number of frames. The original video is first partitioned into segments that have the similar appearance through feature matching. Subsequently, by representing each segment as a node, a sparse graph at each moment is built to indicate the transition between segments. After inferring the adjacency matrix in the sparse time-varying graphs, we are able to detect slide transitions by analyzing its structure.

Video segments. SIFT features are first extracted to represent low-level visual information. With the help of feature matching, we generate a video segment by matching SIFT features in the next few frames to the first frame of current segment until the ratio of feature matches is lower than a threshold m. For instance, if video frame $\#2$ to $\#10$ are similar to frame $\#1$, segment $\#1$ consists of frame $\#1$ to $\#10$. Segment $\#2$ starts from frame $\#11$ and ends at frame whose matches ratio with respect to frame $\#11$ is below the threshold m. To avoid parameter tuning, we adopt an adaptive technique to estimate the threshold m. Specifically, we first compute the ratio of feature matches across the whole video and compute a histogram. By estimating a Gaussian distribution $N(\mu_r, \sigma_r)$ around the peak value with maximum likelihood estimator (MLE), we naturally use $m = \mu_r + 3\sigma_r$ as the threshold to avoid empirically setting. Finally, frame $\boldsymbol{I}^i$ is described as a binary segment indicator vector $\boldsymbol{x}^i \in \Re^D$ with 1 nonzero elements, where D is the number of video segments.

Definition of graphs. The time-varying graph is defined as $\mathcal{G}^i = (\mathcal{V}, \mathcal{E}^i), i \in \{1, \cdots, N-1\}$, where each node in the vertex set $\mathcal{V}$ corresponds to a video segment (*i.e.* $|\mathcal{V}| = D$). The edge set $\mathcal{E}^i$ is encouraged to be sparse and time-varying. On one hand, sparsity is introduced to avoid unnecessary complex graph structure, and the nonzero elements indicates the strong relationship between nodes. On the other hand, $\mathcal{E}^i$ varies smoothly along with the content change over time, which is just used for slide transition detection. The slide transition detection problem is turned to the graph inference one, *i.e.*, how to obtain a set of time-specific adjacency matrix $\boldsymbol{A}^i, i \in \{1, \cdots, N-1\}$ of the edge set $\mathcal{E}^i$, which is detailed in Sec. 4.

Slide transition detection. After obtaining the sparse adjacency matrix, we analyze them to produce the slide transition keyframes. For instance, if a slide stay unchanged, the non-zero elements of matrix $\boldsymbol{A}^i$ always appear in the diagonal position, which means that the node switches to itself. However, during the slide transition period, matrix $\boldsymbol{A}^i$ differs from the previous ones and is not necessarily a diagonal one. Therefore, we are able to detect slide transitions by analyzing the nonzero elements of adjacency matrix. Due to the large amount of frames, we maintain a coarse-to-fine two-step strategy to locate the transition frame efficiently. Specifically, the whole video frames are uniformly split into multiple time intervals (such as every 10 frames), where a coarse-level adjacency matrix is first estimated. A fine-level adjacency matrix is then calculated at each frame to locate the exact time point once a slide transition is found in some certain interval. More importantly, this framework is able to neglect the disturbance such as camera motion and people movement, and reveal the real slide change.

4 Sparse Time-Varying Graphs

In this section, we first describe the graph modeling principles, and then present the optimization framework for solving such sparse time-varying graphs.

4.1 Graph Modeling

The inference of the time-varying graph is formulated as a maximum likelihood estimation problem, based on the assumption that first order Markovian is employed between the consecutive frames. We first describe the graph model in this subsection.

Given a lecture video $\mathcal{I} = \{\boldsymbol{I}^1, \cdots, \boldsymbol{I}^N\}$, after temporal segmenting, we rewrite it as $\mathcal{I} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$. Based on the Markovian assumption, the likelihood of the sequence is defined as

$$f(\mathcal{I}) = f(\boldsymbol{x}^1) \prod_{i=1}^{N-1} f(\boldsymbol{x}^{i+1} | \boldsymbol{x}^i), \tag{1}$$

where $f(\boldsymbol{x}^{i+1}|\boldsymbol{x}^i)$ is the transition model describing the conditional transfer likelihood from frame $\boldsymbol{I}^i$ to frame $\boldsymbol{I}^{i+1}$.

For scalability, we reasonably assume that different video segment x_m^{i+1} and x_n^{i+1}, where $m, n \in \{1, \cdots, D\}$, and $m \neq n$, are conditional independent once given $\boldsymbol{x}^i$. Therefore, the transition likelihood is calculated over each individual dimension

$$f(\boldsymbol{x}^{i+1}|\boldsymbol{x}^i) = \prod_{d=1}^{D} f(x_d^{i+1}|\boldsymbol{x}^i). \tag{2}$$

Naturally, we use a liner dynamic model to simplify the transition model

$$\boldsymbol{x}^{i+1} = \boldsymbol{A}^i \boldsymbol{x}^i + \boldsymbol{\zeta}, \tag{3}$$

where $\boldsymbol{\zeta} \sim N(0, \sigma^2 \boldsymbol{I})$ is a Gaussian noise vector with zero mean and variance σ^2.

Combing Eqn. (2) and (3), the transition likelihood is further expressed as the probability density function of a Gaussian distribution

$$f(x_d^{i+1} | \boldsymbol{x}^i) \sim N(\boldsymbol{A}_{d\cdot}^i \boldsymbol{x}^i, \sigma^2), \tag{4}$$

where $\boldsymbol{A}_{d\cdot}^i$ denotes the d-th row of matrix $\boldsymbol{A}^i$. Taking the logarithm of Eqn. (1), the log-likelihood of the sequence is finally computed as

$$\log f(\mathcal{I}) = \log f(\boldsymbol{x}^1) - \sum_{i=1}^{N-1} \sum_{d=1}^{D} \left\{ \frac{N-1}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} (x_d^{i+1} - \boldsymbol{A}_{d\cdot}^i \boldsymbol{x}^i)^2 \right\}. \tag{5}$$

4.2 Global Optimization

The expected adjacency matrix in the graph should satisfy the following criteria: (1) it should be close to the MLE solution in Eqn. (5); (2) it should only have a few nonzero elements; (3) the matrix in neighboring frames should be temporally coherent. In this subsection we show how to formulate and solve the adjacency matrix in an optimization-based framework.

Firstly, the Maximum Likelihood Estimator in Eqn. (5) produces that $\boldsymbol{x}^{i+1} = \boldsymbol{A}^i \boldsymbol{x}^i$. Due to the insufficient images at time i, the estimator suffer from high variance. Fortunately, supposing that the nearby frames should have the similar appearance, we impose the transition model on the neighboring observation pair $(\boldsymbol{x}^{i+k}, \boldsymbol{x}^{i-k+1}), k \in N^+$ to gather redundant constrict in the data term. Therefore, the first criterion is formulated as

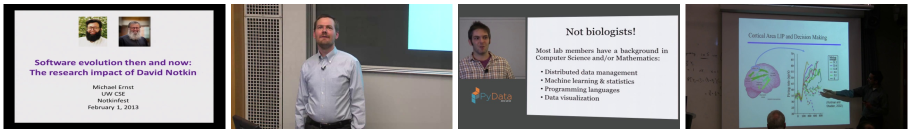
$$\min_{\boldsymbol{A}^i} \sum_{k=1}^{K} w_k^i (\boldsymbol{x}^{i+k} - \boldsymbol{A}^i \boldsymbol{x}^{i-k+1})^2. \tag{6}$$

The weight coefficient w_k^i is introduced to indicate how much degree neighboring frame pair should meet the same adjacency matrix, which is defined as

$$w_k^i = \exp\left\{ -\frac{(2k-1)^2}{2\sigma_t^2} \right\} \exp\left\{ -\frac{||\boldsymbol{x}^{i+k} - \boldsymbol{x}^{i-k+1} - \boldsymbol{x}^{i+k+1} + \boldsymbol{x}^{i-k}||_2^2}{2\sigma_f^2} \right\}. \tag{7}$$

The former part in w_k^t is the temporal weighting of observation pair $(\boldsymbol{x}^{i+k}, \boldsymbol{x}^{i-k+1})$. As time goes on, the relation between observations is gradually weaken. Closer to time i, more contribution on the estimation $\boldsymbol{A}^i$. The latter part in w_k^i is the weighting of segment difference. If the difference between $(\boldsymbol{x}^{i+k} - \boldsymbol{x}^{i-k+1})$ and $(\boldsymbol{x}^{i+k+1} - \boldsymbol{x}^{i-k})$ is large, we think it encounters noise, and set a low weight to avoid the noise. In addition, $\sigma_t = 2$ and $\sigma_f = 0.2$ are standard deviations controlling the Gaussian kernel. $K = \min(\min(N - i - 1, i - 1), 2\sigma_t)$ is the size of the neighborhood set. In addition, this data term is the underlying explanation for being able to discard the disturbance from camera motion and people movement.

Secondly, the graph should only have a few strong connections. This is done by an ℓ_1 regularizer to control the sparsity of adjacency matrix. It not only avoids over-fitting, but also removes the weak link between nodes.


Software evolution then and now:
The research impact of David Notkin
Michael Ernst
UW CSE
Notkinfest
February 1, 2013
Not biologists!
Most lab members have a background in Computer Science and/or Mathematics:
• Distributed data management
• Machine learning & statistics
• Programming languages
• Data visualization
PyData
Cortical Area LIP and Decision Making

(a) Type-A video (b) Type-B video (c) Type-C video

Fig. 1: Three types of lecture videos in our experiments. (a) Type-A video presents complex camera motion and sudden camera switch. (b) Type-B video presents the speaker and computer screen in two regions simultaneously. (c) Type-C video presents the on-stage screen by a single camera.

Thirdly, adjacent frames should have the similar matrix. We minimize the temporal difference $||\boldsymbol{A}^i - \boldsymbol{A}^{i-1}||_2^2$ to maintain temporal coherence.

Finally, we have the complete optimization formula for the adjacency matrix

$$\min_{\{\boldsymbol{A}^i\}} \sum_{i=1}^{N-1} \sum_{k=1}^{K} w_k^i ||\boldsymbol{x}^{i+k} - \boldsymbol{A}^i \boldsymbol{x}^{t-k+1}||_2^2 + \lambda \sum_{i=1}^{N-1} ||\boldsymbol{A}^i||_1 + \alpha \sum_{i=1}^{N-1} ||\boldsymbol{A}^{i+1} - \boldsymbol{A}^i||_2^2, \quad (8)$$

where λ and α are the weights for the sparsity term and smoothness term, respectively ($\lambda = 0.05$, and $\alpha = 0.01$ in our system). This optimization formula is significantly different from that in [8]. On one hand, in [8] each $\boldsymbol{A}^i$ suggests the common codeword transition probability of many photo streams, while we only have a single video sequence. On the other hand, they solve each $\boldsymbol{A}^i$ independently, while we employ a global optimization.

Global optimization in Eqn. (8) could be solved via a lot of tools, such as coordinate descent [3]. Note that the inference of graph reduce to a weighted ℓ_1-regularized least square problem when all variables but $\boldsymbol{A}^i$ fixed. Furthermore, thanks to the assumption that each dimension of $\boldsymbol{x}^i$ are conditional independent, neighborhood selection [11] is applied to obtain each row of $\boldsymbol{A}^i$ separately. As a result, we iteratively solve the weighted lasso problem for each row of the adjacency matrix D times.

5 Results and Discussion

As shown in Fig. 1, we have collected three types of lecture video from Yale University Courses and YouTube to verify the effectiveness and superiority of our system. Type-A video is recorded with multiple cameras which allows complex camera motion and sudden camera switch, such as the switch from slide to speaker. Type-B video is also recorded with multiple cameras but shows the slide and speaker simultaneously. Type-C video captures the speaker and on-stage screen by a still camera. Both camera movement and people movement would affect the detection accuracy. Each lecture video is temporally down-sampled to 1fps and its spatial resolution is 640×360. Video length ranges from roughly 10 minutes to 45 minutes. The parameters are fixed as stated before.

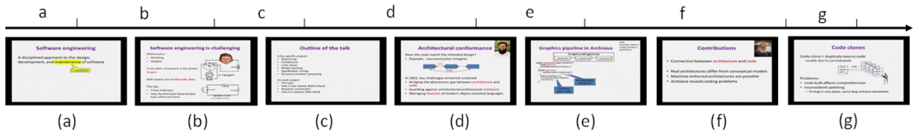


Fig. 2: Slide transition detection result by our approach for Type-A lecture video. Ticks marked on the timeline indicate the detected transition frames shown below the timeline.

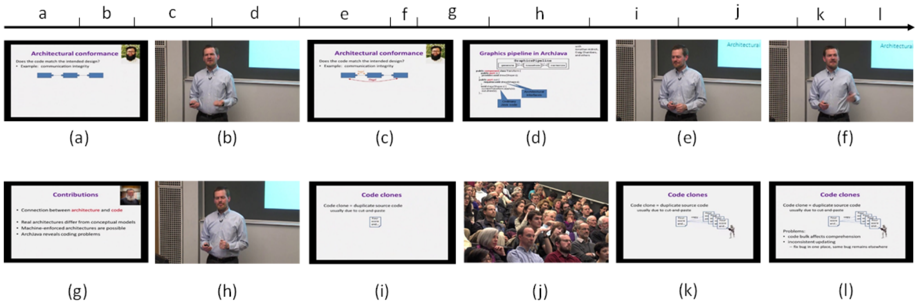


Fig. 3: Slide transition detection result by the SPD approach [9] for Type-A lecture video. Note that both camera switch and people movement are mistaken as slide transitions.

A typical detection result for Type-A lecture video is shown in Fig. 2, where slide progressions are automatically detected and marked on the timeline. Result shows that our method successfully pick out the slide content changes effectively.

We perform some quantitative evaluation to demonstrate the superiority of our system. After manually labeling the groudtruth slide change, we leverage the F_1 score as the evaluation metric

$$\begin{aligned} F_1 &= \frac{2 \times Precision \times Recall}{Precision + Recall} \\ Precision &= \frac{S_c}{S_t} \\ Recall &= \frac{Sc}{S_a} \end{aligned} \quad , \tag{9}$$

Where S_c is the number of slides transitions correctly detected, S_t is the total number of detected slide transitions, and S_a is the total number of the actual slide transitions.

We compare our system with video summarization approach using Singular Value Decomposition (SVD) [4], shot boundary detection method using Frame Transition Parameters (FTP) [12], and recent slide progression detection method by analyzing the feature trajectories (SPD) [9]. Table 1 shows the average Precision, Recall, and F_1

Table 1: Average performance of different methods on three types of lecture video.

	Precision	Recall	F_1 score
SVD [4]	0.997	0.439	0.571
FTP [12]	0.556	0.148	0.215
SPD [9]	0.654	0.947	0.738
Ours	0.850	0.967	**0.904**

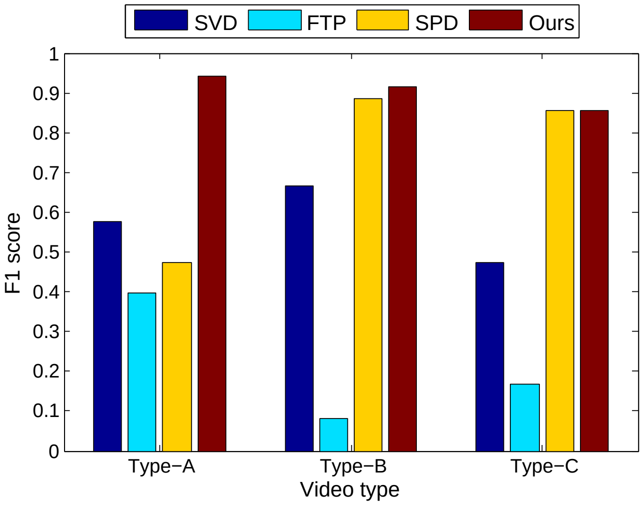


Fig. 4: Detailed performance of different methods on three types of lecture video.

score of different approaches on all types of lecture video, while detailed performance on each type of lecture video is shown in Fig. 4.

As shown in Table 1 and Fig. 4, our system significantly outperforms these approaches in detecting slide transitions. Our system improves the F_1 score by 16.6% on average, compared with the feature trajectory-based SPD approach. In particular, our approach improves the F_1 score up to 46.8% on Type-A lecture video, where camera switch and complex motion are presented. The SPD approach would produce lots of false positives due to the sudden camera switch and frequent people movement when dealing with Type-A lecture video, which is also evidenced by Fig. 3. We also find that general SVD approach achieves a better precision but fails to discover most of the slide changes. In addition, shot boundary detection method using FTP achieves the worst performance for detecting slide transitions.

6 Conclusion

In this paper, we present the sparse time-varying graph optimization approach to automatically detect slides transitions in lecture videos. By formulating the sparsity and time-varying characteristics into a global optimization framework, we are able to solve the adjacency matrix in the graphs and then detect the slide changes. Experimental results show that our system successfully summarizes lecture video by key frames and achieves the best performance. Besides the general feature matching, other specific information of lecture video, such as text title on the top of screen, could be used to further improve the performance, which will be our future work.

References

1. Fan, Q., Barnard, K., Amir, A., Efrat, A.: Robust spatiotemporal matching of electronic slides to presentation videos. IEEE Transactions on Image Processing 20(8), 2315–2328 (2011)
2. Fan, Q., Barnard, K., Amir, A., Efrat, A., Lin, M.: Matching slides to presentation videos using sift and scene background matching. In: Proc. ACM International Workshop on Multimedia Information Retrieval. pp. 239–248 (2006)
3. Fu, W.J.: Penalized regressions: the bridge versus the lasso. Journal of Computational and Graphical Statistics 7(3), 397–416 (1998)
4. Gong, Y., Liu, X.: Video summarization using singular value decomposition. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 174–180 (2000)
5. He, L., Sanocki, E., Gupta, A., Grudin, J.: Auto-summarization of audio-video presentations. In: Proc. ACM International Conference on Multimedia. pp. 489–498 (1999)
6. Jeong, H.J., Kim, T.E., Kim, H.G., Kim, M.H.: Automatic detection of slide transitions in lecture videos. Multimedia Tools and Applications 74(18), 7537–7554 (2015)
7. Kim, G., Sigal, L., Xing, E.P.: Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 4225–4232 (2014)
8. Kim, G., Xing, E.P.: Reconstructing storyline graphs for image recommendation from web community photos. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. pp. 3882–3889 (2014)
9. Li, K., Wang, J., Wang, H., Dai, Q.: Structuring lecture videos by automatic projection screen localization and analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 37(6), 1233–1246 (2015)
10. Ma, D., Agam, G.: Lecture video segmentation and indexing. In: Proc. SPIE 8297, Document Recognition and Retrieval XIX. pp. 82970V–82970V–8 (2012)
11. Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the lasso. Annals of Statistics 34(3), 1436–1462 (2006)
12. Mohanta, P.P., Saha, S.K., Chanda, B.: A model-based shot boundary detection technique using frame transition parameters. IEEE Trans. Multimedia 14(1), 223–233 (2012)
13. Mohanta, P.P., Saha, S.K., Chanda, B.: A model-based shot boundary detection technique using frame transition parameters. IEEE Transactions on Multimedia 14(1), 223–233 (2012)
14. Mukhopadhyay, S., Smith, B.: Passive capture and structuring of lectures. In: Proc. ACM International Conference on Multimedia. pp. 477–487 (1999)
15. Ngo, C.W., Ma, Y.F., Zhang, H.J.: Video summarization and scene detection by graph modeling. IEEE Transactions on Circuits and Systems for Video Technology 15(2), 296–305 (2005)
16. Ngo, C.W., Pong, T.C., Huang, T.S.: Detection of slide transition for topic indexing. In: Proc. IEEE International Conference on Multimedia and Expo. pp. 533–536 (2002)
17. Porter, S.V.: Video segmentation and indexing using motion estimation. Ph.D. thesis, University of Bristol (2004)
18. Repp, S., Meinel, M.: Semantic indexing for recorded educational lecture videos. In: Proc. IEEE International Conference on Pervasive Computing and Communications Workshops. pp. 240–245 (2006)

19. Sujatha, C., Mudenagudi, U.: A study on keyframe extraction methods for video summary. In: Proc. International Conference on Computational Intelligence and Communication Networks. pp. 73–77 (2011)
20. Wang, F., Ngo, C.W., Pong, T.C.: Synchronization of lecture videos and electronic slides by video text analysis. In: Proc. ACM International Conference on Multimedia. pp. 315–318 (2003)
21. Yuan, J., Li, J., Lin, F., Zhang, B.: A unified shot boundary detection framework based on graph partition model. In: Proc. ACM International Conference on Multimedia. pp. 539–542 (2005)
22. Zhao, Z., Cai, A.: Shot boundary detection algorithm in compressed domain based on adaboost and fuzzy theory. Proc. International Conference on Advances in Natural Computation pp. 617–626 (2006)