GRAPH-BASED MULTI-MODAL SCENE DETECTION FOR MOVIE AND TELEPLAY

Su Xu[1], *Bailan Feng*[1], *Peng Ding*[1] *and Bo Xu*[1,2]

[1]Digital Content Technology Research Center, Institute of Automation
[2]National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing 100190, China
{suxu, blfeng, pding, xubo}@hitic.ia.ac.cn

ABSTRACT

Automatic scene detection is a fundamental step for efficient video searching and browsing. This paper presents our current work on scene detection that integrates three effective strategies into a single framework. For each video, firstly, a coherence signal is constructed by graph modal obtained from the similarity matrix in a temporal interval. Secondly, the signal is optimized by scene transition graph (STG) analysis and audio classification, in which scene clues hidden in multimedia are discovered from the video. Finally, the scene boundaries are identified by window function. In experiments, we compare the proposed scene detection method with three typical algorithms on teleplay and movies, and the results of our method, yielding an average 0.85 F-measure, is the best one.

Index Terms— graph-modal, multi-modal, STG analysis, audio classify

1. INTRODUCTION

It is a common sense that automatic technology of scene segmentation provides the most efficient solution to indexing, retrieval and analysis for movie and teleplay [1]. The structure of scene hidden in their plot is difficult to obtain, so researchers have been looking for the effective technology of scene detection.

Several approaches have been proposed for the scene segmentation problem. In [2] and [3], authors first group shots into clusters, and scene segmentation is transformed into a graph partitioning problem. In [4], the normalized cuts method is applied to partition scene. In [5], the authors propose a method of sequence alignment instead of graph partitioning problem. Clustering is an important step in above algorithms, but its stop condition has an adverse effect. The loose stop condition leads to shots are grouped into one cluster from different scenes, caused missing detection; on the contrary, the rigorous one possibly produce a large number of classes, caused false detection. In [6], visual and audio features are both used to scene detection, and scene is modeled by HMMs. Because of the absence of prior knowledge about scenes states, it is difficult to determine the number of states in HMMs model. In [7], the author segment video scene using Markov chain Monte Carlo (MCMC), but its computational complexity is very high. In [8], a video coherence signal is calculated by a graph model, and then a k-means scheme classifies shots into two groups: scene boundary and no scene boundary. However, due to some abnormal shots in the video, this method has lots of false detection.

Scene detection is considered a process to search for reasonable connection between shots in a video. When two shots are not connected by some coherent clues, which clues exists in many aspects, there could be a scene segment point. In this paper, we propose a novel method that utilizes three clues for scene detection, which are visual coherence signal, scene transition graph (STG) and audio type respectively. Because this scheme reveals internal link of a scene in different aspect, it can effectively detect video scene. The rest of the paper will introduce our algorithm in detail and provide sufficient experiments to prove effectiveness of it.

2. THE SCENE DETECTION ALGORITHM

2.1. The Flow of Scene Detection Algorithm

According to the flow chart shown in Fig.1, the video is first divided into shots, and visual and then audio features are extracted based on shots unit. Employing a graph model, the visual coherence signal is calculated to represent shots difference. A high value of the signal implies that there is obvious visual difference between shots, where could occur scene change, but not all of such points cause scene change. Therefore, two strategies are employed to filet the signal. One strategy constructs STG [2] using minimum spanning tree (MST) clustering [9], and the other utilizes clue of sound type by audio classified [10]. After above steps, the scene boundaries can be detected with a window function via identifying local maximum points.

The work was supported by ”The National Key Technology R&D Program”, grant No. 2009BAH40B03.

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

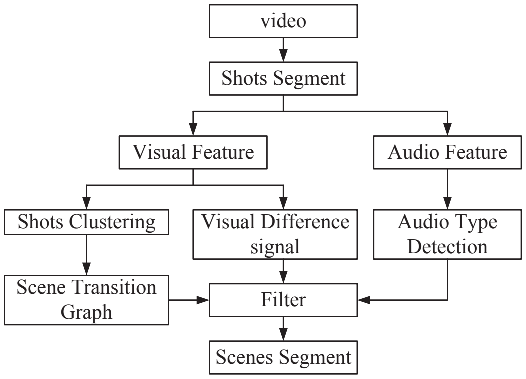


Fig. 1. Flow chart of scene detection algorithm

2.2. Visual Coherence Signal

Selecting key-frames to represent a shot is beneficial to compute visual difference between shots. In order to reducing computational complexity, we use a plain sampling strategy for the key-frame selecting. Assuming sampling step is α, and n_s is the number of frames in a shot. When $n_s > 3\alpha$ key-frames are sampled by step α in a shot, on the contrary, key-frames are only selected by the first, the middle and the last one.

To be robust to noise, the metric employed in [8] is used to compute the distance between two key-frames. As shown in Fig.2, the two key-frames are divided into 16 regions of the same size. We extract a 48-bin RGB normalized color histogram for each region with 16 bins in every color space. Distance of corresponding region is calculated as follows:

$$d = 1 - \sum_{i=1}^{48} \min(H_m^i, H_n^i) \tag{1}$$

Eight regions with the largest differences are discarded to reduce the effects of object motion and noise in Fig. 2. The distance D_k between two key frames is defined as the mean of the histogram differences of the remaining regions. Shot distance is defined as the minimum distance between two groups of key frames form different shots, that is

$$D_s(i,j) = \min\left(D_k\left(i,j\right)\right) \quad i \in m, \;\; j \in n \tag{2}$$

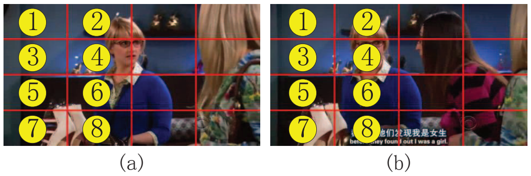
where D_s is shot distance; i and j are index of key frame from different shot; m and n are the number of shots.

The principle of computing visual coherence signal can be explained as min-max cut [8]. The graph $G(V,E)$ is partitioned into two disjoint sets A and B, $A \cap B = V, A \cup B = \phi$, min-max cut criterion is defined as follow:

$$Mcut(A,B) = \frac{cut(A,B)}{assoc(A)} + \frac{cut(A,B)}{assoc(B)} \tag{3}$$

$cut(A,B)$ and $assoc(A)$ are defined as follows:

$$cut(A,B) = \sum_{i \in A, j \in B} edge(i,j) \tag{4}$$



(a) (b)

Fig. 2. Eight regions of the two key frames with similar histograms

$$assoc(A) = \sum_{i,j \in A} edge(i,j) \tag{5}$$

If $2l$ consecutive shots of a video is considered vertices in $G(V,E)$, the signal of min-max cut is calculated as follows:

$$\begin{aligned} &score(i) = Mcut(A,B) = \\ &Mcut\left\{\{S_{i-l}, \ldots, S_{i-1}\}\{S_i, \ldots, S_{i+l-1}\}\right\} \end{aligned} \tag{6}$$

Integer i is an index of shot S_i that between $[i-l, i+l-1]$. The first l shots are sets A, and the last l shots are sets B. Edge of graph is $D_s(i,j)$. Graph model takes shot distance of local neighborhood into account, so it produces a signal with local invariance. After calculating signal, a masking filter acts on it, the formula is

$$score(i) = \begin{cases} \frac{score(i) - m_{score}}{score(i)} & if\, score(i) > m_{score} \\ 0 & if\, score(i) \leq m_{score} \end{cases} \tag{7}$$

where m_{score} is signal median.

2.3. STG Analysis and MST Clustering

A STG analysis can help to get better performance on visual coherence signal. Firstly, we group shots using a MST clustering algorithm [9], which easily add time-constrained in clustering process. For the MST clustering, N shots of a video can be treated as the vertices of a no-oriented graph $G(V,E)$, and weights of those edges are defined based on the distance of shots. All of these edges build a distance matrix $A_{N \times N}$, so an element $a(i,j)$ in the matrix is expressed as follow:

$$a(i,j) = \begin{cases} D_s(i,j) & if \;\; |i-j| < \sigma \\ 1 & if \;\; |i-j| \geq \sigma \end{cases} \tag{8}$$

If temporal distance shots i and j is larger than threshold σ, two shots must belong to different scenes, as well as $a(i,j) = 1$. According to distance matrix, object clusters can be grouped through the following steps:

1)Construct the minimum spanning tree of no-oriented graph $G(V,E)$ by matrix $A_{N \times N}$.

2)Cut the edges whose weights exceed a threshold γ in the MST to form a forest.

3)Find all the trees contained in the forest and consider each tree as a potential cluster.

The STG analysis constructs a scene by backward searching shots in same cluster [2]. This process divides video into a lot of segments, and the scene boundaries are the subset of segments boundaries. For achieving an ideal recall, a rigorous stop condition of clustering should be selected. Although false detection would be increased, it also be removed by further process.

2.4. Audio Classify

STG analysis can optimize coherence signal, where the effective visual clues do not always exist between shots, so audio information can remedy against absence of visual clues. For reasonable scene change, it always accompanies silence or music. On the contrary, if a speech and noise appear between two shots, they should belong to same scene. According to this characteristic, the coherence signal can be further filtered by audio type.

A helpful audio classify algorithm in [10] is used to detect sound type. Features, same as work [10], are extracted from audio data that span two shots and continue half second in each shot. All sounds are classified into silence, speech, music and noise by two support vector machines (SVM), shown as Fig.3.

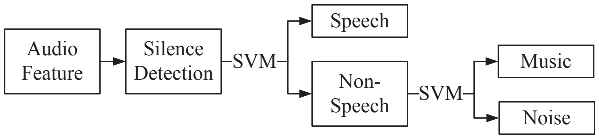


Fig. 3. Flow chart of audio classify

2.5. Scene Boundary Detection

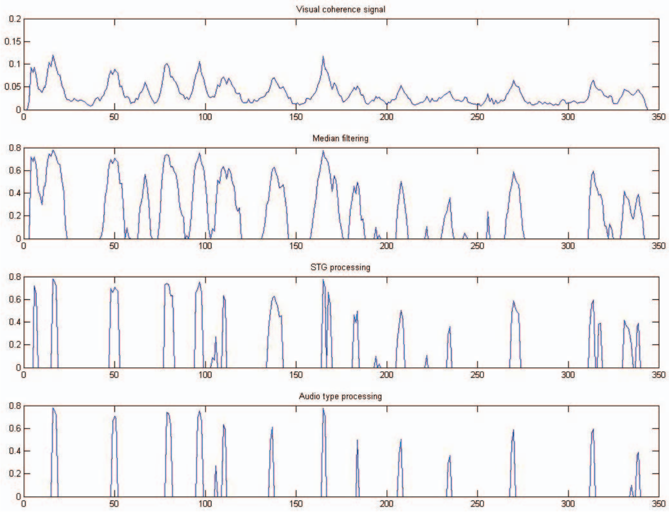


Fig. 4. Result of filtering

The signal calculated in section 2.3 needs further smoothing. Firstly, if a shot located in one segment built by STG analysis, its signal value is set to 0. Secondly, if the audio type between two shots is speech or noise by audio classified, its value also is set to 0. According to results of Fig.4, a lot of false detection can be removed via filtering process.

As shown Fig.4, those local maximum values are scene boundaries, so we can use a window function detecting local maximum values to pick out the boundaries, the formula is

$$\begin{cases} S_i = \max\{S_{i-\delta}, \dots, S_{i-1}, S_i, S_{i+1}, \dots, S_{i+\delta}\} \\ \quad\quad S_i \geq \theta \end{cases} \tag{9}$$

where $2 \times \delta + 1$ is the length of window function. If S_i gets maximum values in the middle of window function and it is larger than threshold θ, S_i is scene boundary.

3. EXPERIMENT RESULTS

3.1. Experimental Setup

To evaluate the performance of our method, we choose three kinds of data, total length about 3 hours, that are sitcom, movie and cartoon for experiment. Table 1 summarizes the information of data set. The data set represents a variety of program genres such as drama or TV sitcom, so the experiments show our algorithm is robust regardless of program genre. For each video, scenes ground-truths are obtained by a human observer in accordance with definition in work [1]. Recall, precision, and F-measure criteria are selected following the work [1] to evaluate the performance of results of experiments.

Table 1. The information of data set.

	Time(min)	Shots	Scene	style
video1	20	351	12	sitcom
video2	21	396	8	sitcom
video3	20	367	11	sitcom
video4	24	395	6	sitcom
video5	31	296	19	movie
video6	30	314	16	movie
video7	22	355	11	cartoon
video8	22	367	10	cartoon
sum total	190	2841	93	

Selecting suitable parameters always depend on the prior knowledge and experience. For our experiments, all the parameters mentioned in section 2 are chosen as follow: $\alpha = 10$, $2l = 10$, $\sigma = 20$, $\gamma = 0.1$, $2 \times \delta + 1 = 13$ and $\theta = 0.3$.

3.2. Results of Scene Detection

With the purpose of making a comparative study to evaluate our method with work [2, 3, 8], we implement all the methods by C++ language and Opencv tools. We also use the same

Table 2. Comparative result with other method using precision, recall and F-measure.

	Our method			Method in [2]			Method in [4]			Method in [8]		
	Prec.	Rec.	Fmea.	Prec.	Rec.	Fmea.	Prec.	Rec.	Fmea.	Prec.	Rec.	Fmea.
video1	0.92	1	0.96	0.47	0.58	0.52	0.47	0.66	0.55	0.57	0.91	0.7
video2	0.8	1	0.89	0.38	0.75	0.5	0.36	0.88	0.51	0.47	1	0.64
video3	0.91	0.91	0.91	0.39	0.63	0.48	0.4	0.72	0.52	0.33	0.63	0.43
video4	0.67	1	0.8	0.38	0.83	0.52	0.38	1	0.55	0.27	0.83	0.41
video5	0.73	0.89	0.8	0.41	0.63	0.5	0.41	0.68	0.51	0.61	0.84	0.71
video6	0.66	1	0.8	0.36	0.56	0.44	0.4	0.75	0.55	0.53	0.93	0.68
video7	0.84	0.84	0.84	0.35	0.54	0.42	0.44	0.72	0.51	0.47	0.81	0.59
video8	0.82	0.9	0.86	0.38	0.6	0.47	0.44	0.8	0.57	0.5	0.8	0.62
Average	0.79	0.94	0.85	0.39	0.64	0.48	0.41	0.77	0.53	0.46	0.84	0.60

visual feature, key frame and shot distance in all the methods, so the core of scene detection can be fairly evaluated. The parameters of work [2, 3, 8] is optimized according to depiction of the papers. The recall, precision and F-measure of the experiments are presented in Table 2. It is clear that our algorithm provides the best results for all videos.

To prove the effectiveness of post-process, we give the results in each step. The final results are optimized by STG and audio type same as our method in Table 2, and the result of coherence signal and STG analysis processing are listed in Table 3. It is obvious that the results are improved in each step. Therefore, we can reasonably conclude that the more clues are used, the better result can be got.

Table 3. Comparative result with different processing.

	coherence signal			STG processing		
	Prec.	Rec.	Fmea.	Prec.	Rec.	Fmea.
video1	0.45	0.75	0.56	0.73	0.91	0.81
video2	0.31	0.75	0.44	0.57	1	0.73
video3	0.33	0.63	0.43	0.53	0.72	0.61
video4	0.21	0.66	0.32	0.5	1	0.67
video5	0.44	0.68	0.53	0.67	0.84	0.75
video6	0.36	0.69	0.47	0.64	1	0.78
video7	0.33	0.72	0.45	0.65	1	0.79
video8	0.32	0.6	0.42	0.5	0.6	0.55
Average	0.34	0.69	0.45	0.6	0.88	0.71

4. CONCLUSION

In this paper, we concentrate on developing a novel method of video scene detection. Three schemes, revealing different clues of video, are integrated into a single framework in order to find scene boundaries. Firstly, coherence signal is obtained using a graph model. Secondly, STG analysis is used to improve performance of coherence signal. Thirdly, audio type is employed to remove false detection. The presented experimental results on several videos indicate that the proposed method accurately detects most scene boundaries, while providing good results on recall, precision and F-measure.

5. REFERENCES

[1] P. Christian “Logical unit and scene detection: a comparative survey,” in *Multimedia Content Access: Algorithms and Systems II*. SPIE, 2008, vol.6820, pp.2–17.

[2] Y. Minerva, Y. Boon-Lock, and L. Bede, “Segmentation of Video by Clustering and Graph Analysis,” *Computer Vision and Image Understanding*, vol.71, pp.94–109, 1998.

[3] C. Ngo, Y. Ma, and H. Zhang “Video summarization and scene detection by graph modeling,” *Circuits and Systems for Video Technology*, vol. 15, pp. 296–305, 2005.

[4] Z. Rasheed, and M. Shah, “Detection and representation of scenes in videos,” *Multimedia*, vol.7, pp.1097–1105, 2005.

[5] V. T. Chasanis, A. C. Likas, and N. P. Galatsanos, “Scene Detection in Videos Using Shot Clustering and Sequence Alignment,” *Multimedia*, vol. 11, pp.89–100, 2009.

[6] J. Huang, Z. Liu, and Y. Wang, “Joint scene classification and segmentation based on hidden Markov model,” *Multimedia*, vol.7, pp.538–550, 2005.

[7] Y. Zhai and M. Shah, “Video scene segmentation using Markov chain Monte Carlo,” *Multimedia*, vol.8, pp.686–697, 2006.

[8] S. Ufuk, and T. Ziya “Video scene detection using graph-based representations,” *Signal Processing: Image Communication*, vol.25, pp.774–783, 2010.

[9] X. Gao, and X. Tang, “Unsupervised video-shot segmentation and model-free anchorperson detection for news video story parsing,” *Circuits and Systems for Video Technology*, vol.12, pp.765-776, 2002.

[10] Y. Li, and C. Dorai, “SVM-based audio classification for instructional video analysis,” in *ICASSP*. IEEE, 2004, vol.5, pp.897–900.