SHOT BOUNDARY DETECTION USING SPECTRAL CLUSTERING

Uros Damnjanovic, Ebroul Izquierdo and Marcin Grzegorzek

Multimedia and Vision Research Group, Queen Mary University of London
Mile End Road, E1 4NS, London, UK
phone: + (44) 20 7882 5346, fax: + (44) 20 7882 7997, email:
{uros.damnjanovic,ebroul.izquierdo,marcin.grzegorzek}@elec.qmul.ac.uk

ABSTRACT

Daily increase in the number of available video material resulted in significant research efforts for development of advanced content management systems. First step towards the semantic based video indexing and retrieval is a detection of elementary video structures. In this paper we present the algorithm for finding shot boundaries by using spectral clustering methods. Assuming that a shot boundary is a global feature of the shot rather then local, this paper introduces the algorithm for scene change detection based on the information from eigenvectors of a similarity matrix. Instead of utilising similarities from consecutive frames, we treat each shot as a cluster of frames. Objective function which is used as the criteria for spectral partitioning sums contributions of every frame to the overall structure of the shot. It is shown in this paper that optimizing this objective function gives proper information about scene change in the video sequence. Experiments showed that obtained scenes can be merged to from clusters with similar content, suitable for video summarisation. Evaluation is done on different datasets, and results are presented and discussed.

1. INTRODUCTION

The pervasiveness of video databases and their growing stocks have attracted significant research to develop tools for effective analysis, structuring, and retrieval of video material. To generate useful multimedia systems, employed tools should link visual information from low level content representation with semantic concepts. Achieving this on a frame level is time consuming and is not feasible for large video databases. Instead of analyzing video on a frame level, it is shown that the better solution is to use one frame that will represent whole shot. Shot detection is a cornerstone for summarisation, syndication and creation of efficient and user friendly semantic annotation interfaces. Though shot detection find application in several other tasks related to computer vision and video analysis, the motivation of this paper has been originated in the problem of semantic based annotation for visual information retrieval.

Shot boundary is a transition between two different scenes containing the same semantic information. Simplest form of the shot change is a cut, abrupt change of the scene, where first frame of the next shot follows the last frame of the previous shot. Detecting cuts is a trivial task, unlike the detection of gradual boundaries. Slow changes are generated by the professional editor, resulting in a change of the visual appearance of frames in a longer period of time. Final result of editing effects can be either change of the shot, or just the change of the visual content inside the same shot. Detecting shot boundaries in a video and clear distinction between various editing effects is still a challenging task. The algorithm that is capable of detecting shot boundaries independently of video type, and editing effects used was the main goal of our work.

According to the features that are used for processing, shot detection algorithms can be classified into uncompressed and compressed domain algorithms. Video segmentation in a compressed domain focuses on the analysis of video features extracted directly from the MPEG compressed stream [1] [2]. Although fast, working in the compressed domain is proven to be less reliable, and less sensitive to different editing effects compared to the algorithms that work with raw video stream. In uncompressed domain information from spatial video domain is used directly. Based on the various visual features a similarity measure between successive pixels is defined. When two consecutive frames are sufficiently dissimilar an abrupt transition is found. For gradual transitions more profound threshold mechanisms are used. In [3] set of pair wise pixel difference metric is defined that measure the content change between video frames. Bloc based approaches use local characteristic to increase robustness to object and camera movements. Each frame is divided into a number of blocks that are compared against their counterparts in the successive frame [4]. In [5] algorithm for detecting the appearance of intensity edges that are distant from edges in the previous frames is described. A method for video shot boundary detect based on the colour histogram differences is given in [6]. The technique uses histogram differences and temporal colour variation to detect different types of boundaries. Algorithm that uses eigenspace decomposition of the RGB colour space for describing the frames in a more descriptive coordinate system is presented in [7]. [8] used interesting approach, the similarity matrix is formed in the similar way like in our method. Shot boundaries are then found by applying the correlation kernel that result in one dimensional vector, indicating if a specific frame is shot boundary or not .

Conventional shot detection methods analyze consecutive frame similarities therefore most of the general information about the shot is lost. Our work is motivated by the assumption that a shot boundary is a global feature of the shot rather then local. General knowledge about the shot is cumulated during the time from the information included in every frame. Information about the boundary is extracted indirectly from the information about the interaction between two consecutive shots. Spectral clustering keeps each frame's contribution in the objective function. By optimizing the objective function when clustering frames, individual shots are separated, and cluster bounds are used for detecting shot boundaries. Using all available data describing the shot, improves the performance of the boundary detection algorithm. By keeping the information in the objective function every frame of the shot has its contribution to the overall knowledge about the boundary. As frames that belong to the same shot are temporally aligned, cluster borders will be points on the time axis. In our algorithm spectral clustering algorithm Normalized Cut [10] is used for clustering. Specially created heuristics is applied to analyze results of the clustering and give final results. Clustering the whole video based on the visual similarity of frames exhibit another interesting property. Resulting clusters will have visual similar frames no matter if they are temporally adjacent or not. This feature can be used to automatically create video summaries.

This paper is organized as follows. Section 2 spectral clustering is summarized while in section 3 the shot detection algorithm based on normalized cuts is explained. Evaluation results are presented in section 4, while conclusion is given in section 5.

2. SPECTRAL CLUSTERING

Spectral clustering has its origin in spectral graph partitioning methods which are popular in high performance computing. Nowadays spectral clustering applies to several areas including machine learning, exploratory data analysis, computer vision and speech processing. In spectral clustering data is a set of similarities w_{ij}, between pairs of points i and j in a dataset V, satisfying $w_{ij}>0$. The matrix $W=[w_{ij}]_{i,j\in V}$ is called the similarity matrix. The problem of bipartitioning the dataset consists of finding the optimal cut between two clusters A and B, where the cut is defined as:

$$cut(A,B)=\sum_{i\in A,\, j\in B} w_{i,j} \qquad (1)$$

Cornerstone of all spectral methods is the optimization of the objective function that describes interaction between partitions. Normalized cut *(Ncut)* criterion is a graph theoretical criterion for splitting the dataset by minimizing the following objective function:

$$Ncut(A,B)=\frac{cut(A,B)}{assoc(A,V)}+\frac{cut(A,B)}{assoc(B,V)} \qquad (2)$$

where $assoc(A,V)=\sum_{i\in A,\, j\in V} w_{i,j}$ is a total connection from points in the partition A to all points of the dataset and

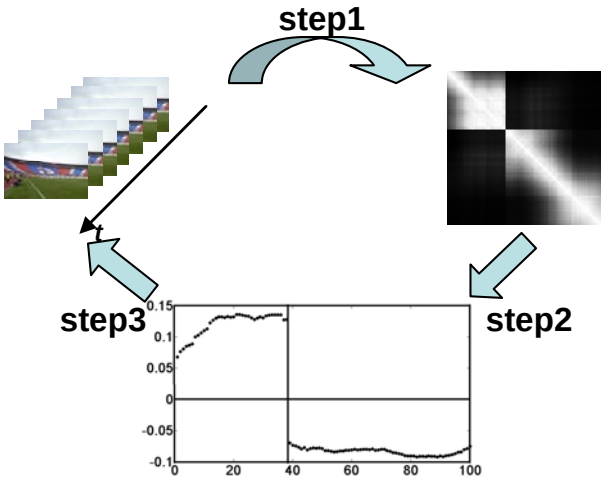

Figure1. Diagram of the spectral clustering algorithm. Step1: pair wise similarity measure is used to map the underlying structure of the dataset to the similarity matrix. Step2: second smallest eigenvector with nearly piece wise constant values reveals the block based structure of the similarity matrix. Step3: information from the eigenvector is used to bipartition the dataset.

$assoc(B,V)$ is similarly defined. The objective function in (2) takes as inputs two clusters of the dataset. In the case of video segmentation, two clusters are defined with one point on the time axis. Thus, objective function can be seen as the function of time:

$$Ncut(t)=Ncut(A,B) \qquad (3)$$

Where A and B are two segments divided by time point t. Shi and Malik [10] showed that optimizing (2) is NP hard. The *Ncut* algorithm is introduced as the approximation method for solving the minimum *Ncut* problem. Minimizing (2) is equivalent to solving the generalized eigenvalue problem:

$$(D-W)x=\lambda Dy \qquad (4)$$

D is $N\times N$ diagonal matrix with $D_{ii}=\sum_i w_{ij}$. Eigenvectors x of (3) are used to bipartition of the dataset. Membership of each point to the specific cluster is indicated by entries of the eigenvector. Partition of the dataset that optimize (2) is indicated with entries of the eigenvector x. Typical spectral clustering algorithm diagram is shown in Fig1. Proper clustering is obtained by assigning each point of the dataset to a specific partition as indicated by the eigenvector. Depending on the algorithm one or more eigenvectors is used to detect underlying structure of the dataset.

3. SHOT DETECTION USING NCUT

As shown in figure1, creation of the similarity matrix is the starting point of every spectral clustering algorithm.

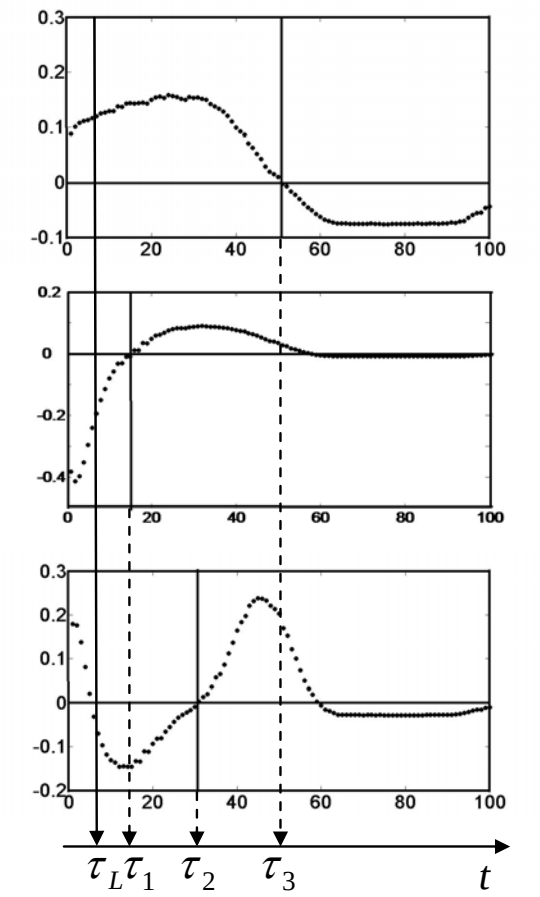

Figure2. Similarity matrix eigenevctors used for finding the shot boundary candidates. Vertical axes correspond to eigenvector entries, showing each frame position relative to the boundary while each frame correspond to a point on the horizontal axis. Possible boundaries are points on the time axis, where the sign of the eigenvector is changed for the first time. τ_L is the starting point of the analysis. τ_1, τ_2 and τ_3 are candidates for shot boundaries.

MPEG-7 colour layout descriptor is used for the low level representation frames. This descriptor is obtained by applying the DCT transformation on a 2-D array of local representative colours in Y, Cm or Cr colour space [11]. Every frame *i* is represented by feature vector F_i with 58 entries. Similarity between frames is calculated by:

$$w_{ij} = e^{-\frac{d(F_i,F_j)}{\sigma^2}} \quad (5)$$

Where $d(F_1, F_2)$ is the distance between two vectors as defined in [11]. Calculating frame by frame similarities for the whole video would take $O(N^3)$ operations, where *N* is the number of frames. The number of operations can be reduced having in mind that frames that are far away in the temporal axis probably don't belong to the same shot. Specifically, we are using sliding window of 100 frames (100<<*N*), where the clustering is carried through. By using the sliding window, number of operations needed to analyze whole video is reduced from $O(N^3)$ to $O(N$) operations. Three eigenvectors corresponding to the second, third and fourth smallest egienvalues are used for describing the structure of video inside the window, Fig2. Every eigenvector indicate possible choice of the shot boundary: τ_1, τ_2 and τ_3. Candidates are chosen to be points on the time axis where values of the eigenvectors change sign for the first time, figure2. The point τ_L is the starting point where the search for candidates starts, and is the minimal number of frames one shot can have. Constraint on the number of frames in a shot is necessary as the quality of spectral clustering depends on the size of clusters. Meila et al. showed that if the similarity matrix is block stochastic, spectral clustering will give the perfect clustering, or in our case will find shot boundaries [12]. In the simplest case, for cuts, the similarity matrix fulfil this condition so eigenvectors will be piece wise constant. Consequently position of the shot boundary would exactly match the position smallest shot candidate, τ_1. In the case of the gradual change, the similarity matrix will have a structure that is more or less similar to the block stochastic matrix, depending on the level of change in frames. In order to detect and classify these changes properly further analysis of eigenvectors is necessary. Define:

$$Ncut_{Li} = f(Ncut(\tau_i)), i = 1,2,3 \quad (6)$$

The objective function threshold value $Ncut_{iL}$ depends on the *Ncut* value in the point τ_1. Similarly, threshold values for the time difference τ_{iL} of the candidates depends on the actual distance between possible boundaries:

$$\tau_{Li} = g(\tau_{i+1} - \tau_i), i = 1,2 \quad (7)$$

Reasoning algorithm for the analysis of eigenvectors is as follows:

Input: Eigenvectors cut points τ_1, τ_2 , τ_3 with respective *Ncut* values in these points.

Output: Single frame candidate for shot boundary τ_B

1. Initialize frame candidate: $\tau_B = \tau_L$.
2. **If** $Ncut(\tau_1) < Ncut_{L1}$
3. **If** $(\tau_2 - \tau_1) > \tau_{L1}$ initialize output: $\tau_B = \tau_1$
4. **ELSE (3)**
5. **If** $Ncut(\tau_1) < Ncut(\tau_2)$ initialize output: $\tau_B = \tau_1$
6. **ELSE (5)**
7. **If** $Ncut(\tau_2) < Ncut_{L2}$
8. **If** $(\tau_3 - \tau_2) > \tau_{L2}$ initialize output: $\tau_B = \tau_2$
9. **ELSE (9)**
10. **If** $Ncut(\tau_2) < Ncut(\tau_3)$ initialize output $\tau_B = \tau_2$
11. **ELSE (10)**
12. **If** $Ncut(\tau_3) < Ncut_{L3}$ initialize output: $\tau_B = \tau_3$
13. **ELSE (12)** there is no boundary in the window
14. **ELSE (7)** go to the step 10.
15. **ELSE (2)** go to the step 7

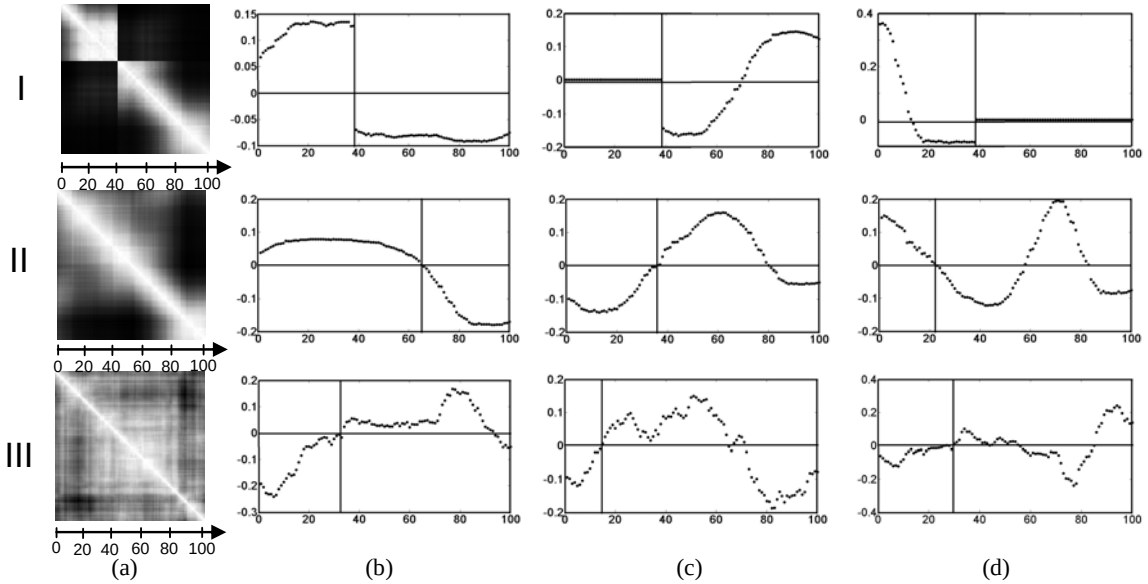

Figure3. Three typical examples found in our experiments. (a) Similarity matrix. (b)- (d) Second, third and fourth eigenvectors providing shot boundary candidates. Horizontal axis corresponds to single frames (1-100) inside the sliding window. Respective eigenvector values are shown on the vertical axis. First sign change in the eigenvector values indicates the shot boundary candidate. Row I: Abrupt change of shots, similarity matrix has the clear block structure. All eigenvectors give the same frame for cut candidate. Row II: Gradual change of shots, block structure of the similarity matrix is not so clear like in the cut case. Every eigenvector gives different candidate for shot boundary. Row III: No shot change within the sliding window. Similarity matrix consists of frame similarities from the same shot.

Once the shot boundary candidate is found, statistical analysis of the *Ncut* value in the adjacent frames is brought about. First *Ncut* value is calculated for 5 frames before and after the τ_B. Mean value μ_n and standard deviation σ_n of *Ncut* values are found. In [13] is showed that the performance of shot detection methods can be improved by the use of a threshold that adapts itself to the sequence statistics. Adaptive threshold used in our algorithm is defined by:

$$m_T = T_N \mu_N + T_D \sigma_N \quad (6)$$

Adaptive threshold algorithm is applied to the objective function values. If the objective function in point τ_B is less than the threshold defined in (6), τ_B is the shot boundary. If the objective function is greater than the threshold defined in (6), current window contain no shot boundaries. By sliding the window over the time axis, whole video is segmented, and shot boundaries are extracted independently of the category of the change.

4. EXPERIMENTS

For assessment of our algorithm three different video genres have been used for shot boundary detection task. Particular video genres have different characteristic editing effects. Dataset that was used for experiments was carefully chosen to include all categories of shot changes. News videos are used to test the performance of the algorithm in detecting long shots with abrupt changes. 2431 seconds of news videos with total of 388 shot changes is used. Next, total of 1576 seconds of music videos is used with 398 shot changes. In music video there is no rule for modelling the structure of the shots. Fast changes of scenes, extensive use of animation effects, fast camera movements zooming in and are all factors that make the shot detection in music videos hard task. Finally 1875 seconds of documentary videos is used with total of 332 shot changes. Documentaries are rich with gradual changes between shots, intensive camera movements, and changes in the light intensities within the shots. Fig3 shows three typical cases from our experiments. Example of an abrupt change of the shot is shown in the row I. Clear block structure is defined in the similarity matrix Fig3 (a), and all three eigenvectors have the cut in the same frame, figure3 (b-d). Case of a gradual change is shown in figure3, row II. Block structure of the similarity matrix is not so clear in this case which results in three different cuts in each eigenvector. To detect the proper position of the shot change algorithm for eigenvectors analysis is employed. Finally the case where there is no boundary in the window is shown in Fig3, row III. All frames within the window have high similarities, and resulting eigenvectors have unclear structure. Eigenvectors are analyzed with our algorithm and no boundary is detected.

By letting the sliding window enclose all frames of the video, and using more then three eigenvectors, our algorithm will cluster shots based on the content similarity. If there are two scenes in the video that are separated in time the technique will cluster them together. This is a useful feature which makes possible more then just shot detection. It also produces video summaries by clustering similar scenes. This is due to the fact that time is not considered in our similarity measure. Leaving out the temporal information from the

analysis, information from eigenvectors can be used to cluster shots based on the visual similarity of the content. The choice of the proper similarity measure is the crucial factor for the clustering performance. Successful shot detection and video summarisation can be accomplished based on how successfully the underlying data structure is embedded in the simailrtiy matrix.

For showing the actual performance of the algorithm standard precision and recall are used. Results obtained with three different video types are shown in table1. TP refers to the number of correctly detected boundaries. FP is the number of boundaries incorrectly detected, while FN is the number of missed boundaries.

Video	TP	FP	FN	Recall [%]	Precision [%]
News	357	20	31	92	94.6
Documentary	314	29	18	94.5	91.5
Music	332	45	66	88	83.4

Table1. Boundary detection results for three different genres of videos.

Generally our algorithm shows good results, with precision and recall over 90%, for documentaries, and news video. Music video showed to be a difficult task. Total editing freedom makes it hard to use some predefined reasoning for eigenvectors analysis. High number if false positive results come from big changes in scenes like change in the light, overall colour of the scene and extensive use of animation effects. This results in a change of the colour appearance of frames, which is reflected in big distance between feature vectors. Even with many factors that are making shot boundary task difficult for music videos, results are reasonable good.

5. CONCLUSION

In this paper we introduced shot detection method that segments the video by using spectral clustering technique. It is described how objective function, optimized by spectral clustering, can save the contribution of each frame to the overall structure of the shot. Performance of the algorithm for spectral clustering of frames into shots together with applied heuristics for eigenvectors analysis is tested on three different video genres. It is shown that performance of algorithm is highly dependent on the choice of the similarity measure. We are investigating ways to improve the similarity measure between frames. By making the similarity matrix less sensitive to changes in the video clustering results will expose the video structure more reliable. We also showed that by extending the domain of clustering from frames to shots, it is possible to cluster shots with similar content into same cluster. This property of our technique can be extended to enable automatic creation of video summaries.

6. ACKNOWLEDGMENTS

This research was supported by the European Commission (FP6-027685-MESH). The expressed content is view of authors but not necessarily the view of the MESH project as whole.

REFERENCES

[1] J. Calic and E. Izquierdo, "Towards real time shot detection in the MPEG compressed domain" in *Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services, 2001*

[2] B. L. Yeo, B. Liu, "Rapid scene analysis on compressed video", *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 5, no. 6, pp. 533-544, Dec. 1995.

[3] H. Zhang, A. Kankanhalli and W. Smoliar, "Automatic partitioning of full motion video", *Multimedia Systems*, vol.1, no.1, pp. 10-28, Jun 1993

[4] R. Kasturi and R. Jain, "*Computer vision: Principles*", IEEE Computer Society Press, 1991

[5] R.Zabih, J. Miller and K. Mai, "A feature based algorithm for detecting and classifying scene breaks", *Proc. ACM Multimedia* 1995, pp. 189-200.

[6] S. H. Han, K. J. Yoon and I. S. Kweon, "A new technique for shot detection and key frames selection in histogram space", *Image Processing and Image Understanding*, 2000

[7] A Yilamz and M. Shah, "Shot detection using principal component system", *Proceedings of IASTED Internet and Multimedia Systems and Applications Conf. (IMSA), 2000*

[8] M. Cooper, "Video segmentation combining similarity analysis and classification", *Proceedings of the 12th annual ACM International Conference on Multimedia*, 2004 pp. 252- 255

[9] W. Zhao, J. Wang, D. Bhat, K. Sakiewicz, N. Nandhakumar and W. Chang, "Improving colour based video shot detection", *IEEE International Conference on Multimedia Systems*, vol. 2, pp. 752-756, Jul 1999.

[10] J. Shi and J. Malik, "Normalized cuts and image segmentation" *IEEE Transactions on PAMI* vol. 22, No. 8, pp. 888-905, August 2000.

[11] B. S. Manjunath, P Salembier and T. Sikora, *"Introduction to MPEG-7"*, John Willey& Sons, 2002.

[12] M. Meila, S. Shortreed and L. Xu, "Regularized spectral learning", UW Statistics Technical Report 465, 2005.

[13] Y. Yusoff, W. J. Christmas and J. Kittler, "Video shot cut detection using adaptive threshold", *BMVC* 2000

[14] R. Zhao and W. I. Grosky, "A novel shot detection technique using colour anglogram and latent semantic indexing", *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops* 2003, pp. 550-555.

[15] X. Liu and T. Chen, "Shot boundary detection using temporal statistics modelling", *International Conference on Acoustic, Speech and Signal Processing, Proceedings* (ICASSP '02) pp. 3389-3392.