

Scene Boundary Detection with Graph Embedding

Jeong-Woo Son, Wonjoo Park, Minho Han, Sun-Joong Kim

Electronics and Telecommunications Research Institute, Smart Media Platform Lab.

218 Gajeong-ro, Yuseong-gu, Daejeon, 34129, Korea

{jwson, wjpark, mhhan, kimsj}@etri.re.kr

Abstract— Scene boundary detection is a well-known task in both computer vision and machine learning. Due to the different characteristics of scene boundaries according to applied aspects, scene boundary detection can be casted into an unsupervised learning with multi-view data. This paper suggested the scene boundary detection method which adopts several ways to handle information in multi-view data. More specifically, in the proposed method, a shot is represented with multiple features and then their relations are represented with multiple affinity graphs. In this situation, this paper explains how multiple graphs are combined in a single complementary graph without information loss. In experiments, we tested five methods to combine graphs by using six Korean TV-series.

Keywords— Scene boundary detection, clustering multi-view data, graph embedding, spectral clustering, video segmentation

I. INTRODUCTION

Scene boundary detection is a well-known task in both computer vision and machine learning. Since the characteristics of scene boundaries in a video content are varied according to the genre of the content, in most cases, it has been performed with unsupervised manner. Under this problem setting, scene boundary detection can be casted into a clustering task. The goal of the scene boundary detection is to construct clusters composed of several shots. A shot can be represented with various features such as colour, motion, sound, and text from its closed captions. Thus, to achieve the goal of the scene boundary detection, a clustering method for multi-view data is demanded. In this paper, we proposed a scene boundary detection system with graph embedding and combining. As a pre-process step, we segment a video content into shots with colour histograms and extract visual, audio, and text features from each shot. When a shot is represented with various features, we can construct several affinity matrixes for shots. Before generate clusters, we need to combine those affinity matrixes into a single affinity matrix. In this step, we adopt various embedding and combining methods to generate low-dimensional affinity representations for shots. First, as a simple method, we adopt kernel production and summation method. Second, co-training based information propagation and spectral embedding [1] is used. Third, a graph embedding method is implemented with 4-layered (512 nodes-256 nodes-128 nodes-64 nodes) denoising auto-encoder [2]. A denoising auto-encoder is designed to embedding an instance into low dimensional space by minimizing re-construction error. Thus, a graph embedding

can suggest low-dimensional vectors for shots minimizing information loss. For the second and third methods, the final vector representation is constructed by concatenating vectors from multiple views.

Clusters are generated with a k -means clustering. This sequential process, composed with graph embedding and k -means clustering, can be considered as a spectral clustering that performs a spectral embedding and k -means clustering. Each generated cluster can be regarded as a scene. Experimental results are derived with six episodes in two Korean TV-series. In experiments, the performances of the proposed system is reported according to the combining method. A single view based method, kernel summation and production, co-training based approach, and graph embedding are compared. Consequently, experimental methods are compared with respect to their scalability and accuracy.

II. SYSTEM OVERVIEW

Figure 1 shows the overall structure of the proposed system. As shown in this figure, the proposed system is composed of mainly three steps: shot extraction, scene clustering, overlap-link resolving. This figure shows the system with a graph embedding. However, note that it is simple to reflect other combining methods by exchanging the second step. The shot extraction aims to extract a number of shots. The number of shots m should be less than or equal with the number of scenes n ($m \leq n$). In this step, a HS (Hue and Saturation) histogram is used to represent a frame and a cosine similarity is adopted to determine how two adjacent frames are similar. After estimating similarities, the shot boundaries are decided with a threshold θ . That is a similarity between two frames is below the threshold, a shot boundary is determined at this frame.

Extracted shots are represented with multiple aspects. Since shots is a sequence of frames, we can obtain more information from shots than one from a frame such as motions and audio features. Table 1 shows the simple description of used features and their categories. As shown in this table, the system extracts features for shots with seven categories based on twenty five features. After multiple aspects are obtained, a affinity matrix can be constructed with a single category. That is, we can obtained seven affinity graphs from features in Table 1. A Gaussian kernel is applied to determine similarities between shots. The kernel width is estimated with the empirical mean of Euclidian distance between shots. Figure 2 shows an example of affinity graphs that are constructed with

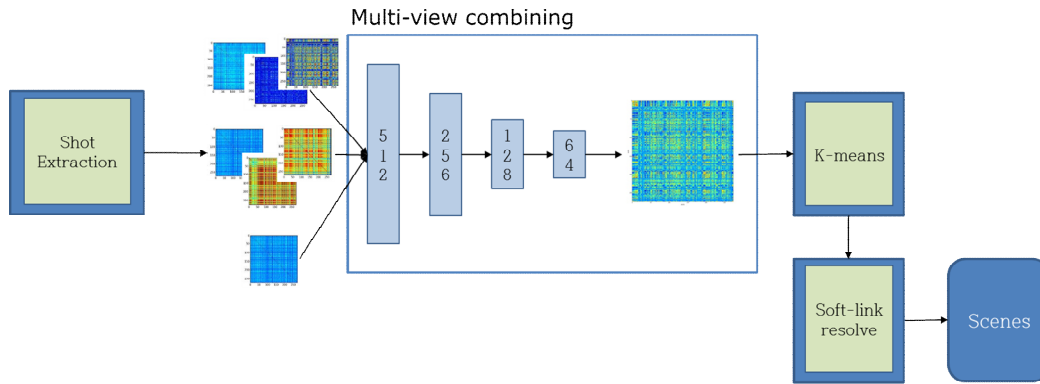


Figure 1. The overall structure of the proposed system

seven aspects. As shown in this figure, each aspect represents a different graph structure from others.

Table 1. Features and their category

Category	Feature Name
Spectral descriptors	Bark/Mel/ERB bands, MFCC, GFCC, spectral peaks, complexity, HFC, dissonance
Tonal descriptors	Pitch salience, chords, ...
Time-domain descriptors	Zero-crossing-rate, loudness, ...
Colour info.	HS histogram, grey intensity
Motion	Motion vector
Others	Hog, AKAZE, ...
Text	Closed caption, manuscript

III. GRAPH EMBEDDING

Multiple aspects represent their own local manifold structures. Thus, to cluster shots with information in multi-view data, it demands to embedding and combining multiple manifold structures into a single manifold by preserving complementary information. For the proposed system, we adopt four methods. Among them, two methods (kernel summation and kernel production) follow the simplest way to make a single vector from multiple vectors. Since the similarities between shots are determined with a Gaussian kernel, it can be regarded as probabilities of a shot given another shot. In this case, kernel summation and kernel production can be considered as a way to determine a conditional probability with independent variables. That is, in both methods, it is assumed the independent among views. Due to the lack of the space, we do not describe detail equations for kernel summation and kernel production, but it can be found in text books [3].

A. Co-training approach for spectral clustering

Multiview spectral co-clustering is a spectral clustering with co-training. The similarity graphs are constructed with each view. Then, graph Laplacians are determined and Eigen vectors are constructed separately. Constructed Eigen vectors are applied to the other view. By applying the Eigen vectors of a single view to the similarity graph of the other view,

information is propagated between views. Table 2 shows the algorithm of the multi-view spectral co-clustering. In this algorithm, $sym()$ is the function that makes the input matrix to be symmetric. While, $concat()$ is a column-wise concatenation of input matrices.

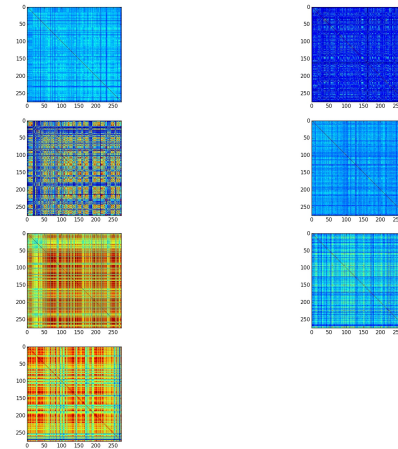


Figure 2. Graphs from multiple aspects

Table 2. The algorithm of multi-view spectral co-clustering

Input: similarity matrixes K_1, K_2 , Max iteration $iter$
Output: k clusters
Init.: Graph Laplacians L_1, L_2 , Eigen vectors U_1^0, U_2^0
for $i = 1$ to $iter$ do
1: $S_1 = sym(U_2^{i-1} U_2^{i-1T} K_1)$
2: $S_2 = sym(U_1^{i-1} U_1^{i-1T} K_2)$
3: compute U_1^i, U_2^i with S_1, S_2
End for
4: normalize U_1^i, U_2^i
5: $V = concat(U_1^i, U_2^i)$
6: perform k -means clustering with V
7: return the clustering result

When more than three views are given, the Multiview spectral co-clustering considers them equally. By assuming independences among views, the 1 and 2 steps in the algorithm in Table 1 is modified as

$$S_v = \text{sym}((\sum_{v' \in \neg V} U_{v'}^{i-1} U_{v'}^{i-1T}) K_v),$$

where v is the target view, while $\neg V$ is the set of other views.

B. Graph embedding with Auto-encoder

Auto-encoder is a typical method in deep learning techniques. An auto-encoder method is composed of certain number of layers. Each layer has k number of nodes that fully connected with a k -dimensional input and generate k' outputs that are $k' \ll k$. When k -dimensional input is given, the outputs of a layer is determined as $z = w \cdot x$ where x and z are the input and embedded vector respectively, while w is a parameter to be estimated. The parameter w is estimated to minimize reconstruction error. That is,

$$w^* = \min_{w \in W} (x - z)^2.$$

The training of multi-layer auto-encoder is performed iteratively for each layer. In the proposed system, we adopted four layered auto-encoder with 50% corruption to prevent over fitting and to robust noise.

IV. EXPERIMENTS

Experiments are performed with six Korean TV-episodes from two TV-series (Heard It through the grapevine and Secret) used in [4]. Table 3 shows experimental results. In this experiment, the performance is measured with the average ARI (Adjusted Rand Index). The graph embedding achieves the best performance, while co-training approach shows the second best. Unexpectedly, both kernel summation and production shows the worst performances. This result shows the effectiveness of view combining methods. With respect to the computational time, co-training approach shows the reasonable results. Even though graph embedding demands much more computational time than others, it still have a strong advantage. When the number of shots is increased, co-training approach cannot handle that since it uses Eigen vectors, while the graph embedding can be applied to a number of shots easily. Thus, the graph embedding is appropriated for the content like movies and documentary.

Table 3. Experimental results

Combining Method	Mean ARI	Time (Sec)
Single view	0.576	0.838
Kernel summation	0.499	0.963
Kernel production	0.416	0.971
Co-training	0.589	8.613
Graph Embedding	0.603	163.679

V. CONCLUSION

This paper proposes a system to detect scene boundaries in video contents. The proposed system uses multiple aspects of shots to improve scene boundary detection performance. To handle such multiple aspects, it provides kernel summation, kernel production, co-training approach, and graph embedding. In experiments, we evaluate those four methods with six Korean TV-episodes. We have a plan to improve performance by using sequential adaptation for chained tasks [5].

ACKNOWLEDGMENT

This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [16ZC2200, Development of programmable interactive media creation service platform based on open scenario]

REFERENCES

- [1] A. Kumar and H. Daumé III, "A Co-training Approach for Multi-view Spectral Clustering," In *Proc. of the 28th ICML*, pp. 393-400, 2011.
- [2] F. Tian, B. Gao, Q. Cui, E. Chen, and T.-Y. Liu, "Learning Deep Representations for Graph Clustering," In *proc. AAAI 2014*, pp. 1293-1299, 2014.
- [3] C. Bishop, "Pattern Recognition and Machine Learning," Springer, 2006.
- [4] J.-W. Son, S.-Y. Lee, S.-Y. Park, and S.-J. Kim, "Video Scene Segmentation based on Multiview Shot Representation," in *Proc. ICTC 2016*, pp. 381-383, 2016.
- [5] J.-W. Son, H. Yoon, S.-B. Park, K. Cho, and W. Ryu, "Consolidation of Subtasks for Target Task in Pipelined NLP Model," *ETRI Journal*, vol. 36, no. 5, pp. 714-720, 2014.



Jeong-Woo Son received his MS and Ph.D. degrees in computer engineering from Kyungpook National University, Daegu, Rep. of Korea in 2007 and 2012 respectively. Since 2013, he has been with ETRI, Daejeon, Rep. of Korea. He focuses on machine learning, NLP, and information retrieval.



Wonjoo Park received her MS degrees in information and communication engineering from Chungnam National University, Daejeon, Rep. of Korea in 2000. She joined ETRI, Rep. of Korea in 2000, where she is currently senior researcher. Her research interests includes data mining, topic model, and ontology.



Minho Han received his BS and MS degree in computer engineering from Chungnam National University, Daejeon, Rep. of Korea in 2001. He joined ETRI, Rep. of Korea in 2000, where he is currently senior researcher. His research interests includes natural language processing and information retrieval.



Sun-Joong Kim received her BS degree in computational statistics and her MS degree in computer science from Chungnam National University, Daejeon, Rep. of Korea, in 1989 and 2000 respectively. In February 1989, she joined ETRI, Daejeon, Rep. of Korea, where she is currently principal researcher and director. Her research interests includes convergence service control, smart TV, content knowledge mining.