

Rule-Based Scene Boundary Detection for Semantic Video Segmentation

Yue Feng, Reede Ren, Joemon Jose
MIR group
Computer Science
University of Glasgow
Glasgow, UK, G12 8QQ
{yuefeng, reede, jj}@dcs.gla.ac.uk

Keywords: Semantic video segmentation, low-level features, high-level rule.

Abstract

In this paper, we present a novel method for semantic video segmentation by using both low-level features and high-level rules on videos and managing it in a hierarchical structure of key-frame, shot and scene. Features in color domain is calculated and utilized for detecting the key-frames and estimating the similarity between shots. By applying the pre-defined high-level rules, similar shots are merged and the scene boundaries are determined. Finally, a likelihood function is designed for improving the accuracy of scene boundary results. Experimental results from several Hollywood movies have demonstrated and show a better performance of both precision and recall has been achieved comparing with other existing works.

1 Introduction

Driven by the bloom of multi-media devices and application, automatic video segmentation has gained numerous attentions due to its commercial potential value in markets. A good video segmentation system can help users to have a well organisation and index of large numbers of video content inside the video library and it is also a bridge linking the raw low-level video data to the high-level semantic fields. A video segmentation system, featured in converting low-level visual features to the high-level, should highlight the video content and structure information including video shots and scenes.

There are numerous techniques in video segmentation, in broad, it generally involves using shot [1~3] or scene-based approaches [4, 5]. The shot-based method first detects different shots in a film and uses few key-frames to represent one shot. One existing work can be found at [6], where a shot detection approach based on fuzzy logical analysis of image histogram is introduced and key-frames with large difference in color domain are extracted for representing the detected shots. The scene-based method is a hierarchical level of structuring videos, which uses scene, shot, and key-frames to represent and construct the whole video, where a video comprises of various scenes and each scene contains a group of shots and every shot is represented by a set of key-frames

extracted from all the frames inside one shot. One example of this existing approach can be found at [7], the author proposed a method using graphical representation to represent videos by constructing a scene transition graph, which uses nodes and edges to represent shots and transitions, respectively. A complete-link method of hierarchical clustering helps to split the video into sub-graphs, each sub-graph will represent a scene.

A scene detection method using motion information as main feature is introduced at [8], where the video structure is built up using a group of spatio-temporal slice. Then, each shot was represented either by detected key-frames or mosaics. However, the motion information here is only for excluding the moving foreground from mosaics and not as a cue for detecting shot similarities. Although, certain level of result has been achieved, this method is only limited to some video with obvious motion information, such as action movies, and therefore proper mosaic construction and matching is difficult to achieve.

Recently, there has been increasing research effort for automatic generation of links between low-level features and high-level concepts in video segmentation, such as using rule-based system [9]. In [10], it reported that the low-level visual features are firstly extracted from video data sets. Then the fuzzy logic and rule mining techniques are applied to approximate human-like reasoning to generate the rules for video segmentation. The results show robustness and high accuracy of the method, however, the process speed of this method is sacrificed by using the fuzzy logic algorithm.

To inherit the merits of the existing approach and improve processing speed, the research leading in this paper is a rule-based hierarchical video segmentation analysis using high-level concepts derived from low-level features. Each video frame is represented by HSV color histogram and histogram matching is applied to extract key-frames, video shot boundaries, and also to help to define the high-level concepts. And then, rule-based scene boundary selection is used to classify the detected shots into different scenes, following which a likelihood-based scheme is also introduced to improve the accuracy of the detected scenes boundary.

The rest of this paper is organized as follows. Section 2 describes the detailed design of the proposed algorithm for

video segmentation, and experimental results and conclusions are illustrated in Section 3 and Section 4, respectively.

2 Scene Boundary Detection

A video usually contains a group of scenes, where each scene includes one or more shots, while every single shot is an uninterrupted segment of video frames taken by one camera. A group of shots filmed in the same place is then built up into one scene, which generates a story topic. Finally, a full film is gathered by various short scenes.

In order to manage the massive video contents in an efficient way, we adopted the method in [6] to temporally partition video into shots, which uses two steps to detect abrupt and gradual shot cuts, (i) A combination of representative features, such like color, motion, edge and texture, is first designed to represent each frame; (ii) Fuzzy logic approach is utilized to find shot boundaries, where the designed feature has a peak change on the shot boundary. More details of this method can be found in [6]. Given the extracted shot boundaries, key-frames for each shot are extracted using color features, and then shot similarity is computed using the color similarity of key-frames from each shot. Finally, scene boundaries for the whole video are obtained by estimating the shot similarities and using a set of pre-designed rules.

2.1 Extracting Key-Frames

With the detected shots from the whole film, a high-level feature extraction for structuring the video is needed to collect the most characterized information rather than using all video frames to avoid the costly computation and speed up the system.

After reviewing the existing methods in literature [7, 12, 13], where the motion, texture, and color feature is applied, it seems that the color histogram is the most suitable candidate for key-frame extraction due to its unique character of non-sensitivity to motion and good performance in representing video contents. Specially, any visual changes in frames will directly lead to the change in the color histogram. The color histogram applied in this paper is a normalized 16-bin HSV color feature, which uses eight bins for hue, four bins for saturation and four bins for value. Thus, each frame can be represented as a color histogram.

Given i detected shots, the proposed key-frame extraction can be achieved in the following steps, firstly the first frame f_1 of each shot is always chosen as the first key-frame and stored in the key-frame set. Color similarity between the next coming frame and the key-frames in the key-frame set is computed. If the calculated color similarity is below threshold, T_1 , this coming frame will be regarded as a key-frame as well. Otherwise, just input the next following frame. A full key-frame set can be estimated after looping through all the remaining frames inside the same shot.

This key-frame extraction method can also be represented as pseudo-code.

```

for  $\forall$  shots  $S_i \in Video$  // where  $i$  is the number of
                        detected shots inside one video
    key-frame set =  $\{f_1\}$ 
    for ( $f_j, 2 < j < N_j$ ) // where  $j = 2$  to  $N_j$ ,  $N_j$  is the
                        number of frames in  $S_i$ 
        if  $ColSim(f_j, key-frame set) < T_1$ 
            key-frame set =  $\{f_1, f_j\}$ 
        end
    end
end

```

The pre-designed threshold, T_1 , is decided not only for selecting the most characteristic frames as key-frames to represent the whole shot but also to reduce the redundancy of contents inside of the shot and to speed up the process speed.



Figure 1. Key-frames extracted from the 1st detected of 'Ace Age 2'

The color similarity is designed to represent the degree of the similarity between frames, and it can be formulated as:

$$ColSim(x, y) = \sum_{h \in bins} \min(H_x(h), H_y(h)) \quad (1)$$

where H_x and H_y are the histograms of frames x and y , respectively. $ColSim(x, y)$ lies between $[0, 1]$ representing the similarity of two video frames. This ensures that the redundancy of the video is reduced due to key-frames are used and capable to represent the shot instead of using all the frames in a shot. More details of this color similarity calculation method can be found at [5]. Figure 1 shows the extracted key-frames of one shot from 'Ace Age 2'.

2.2 Determining Similarities between Shots

Obviously, the shots in one scene should have similar visual contents with reasonable content meaning to make the film feel successive and understandable. Thus, a HSV color histogram feature is used to estimate the similarity between

two shots, $ShotSim(G, K)$, and it is computed as the Hausdorff distance [14], the key-frame set $G = g_1, g_2, \dots, g_{n1}$ and $K = k_1, k_2, \dots, k_{n2}$, which can be formulated as:

$$ShotSim(G, K) = H(G, K) \quad (2)$$

$$H(G, K) = \max(h(G, K), h(K, G)) \quad (3)$$

$$h(G, K) = \max_{g \in G} \min_{k \in K} (g, k) = \max_{g \in G} \min_{k \in K} \|g - k\| \quad (4)$$

$$h(K, G) = \max_{k \in K} \min_{g \in G} (k, g) = \max_{k \in K} \min_{g \in G} \|k - g\| \quad (5)$$

The value of $ShotSim$ lies between $[0, 1]$, the more two shots share the similar visual contents, the higher value of $ShotSim$ it achieves. This method assures that the amount of difference between shots under the HSV color field can be calculated using minimum processing cost.

2.3 Rule-based Estimation of Scene Boundaries

After parsing the video into shots, extracting representative set of key-frames for each shot, and finding shot similarity between shots, the next step for video segmentation is to find the scene boundaries.

Rule-based techniques are known for their effective grouping of semantic concepts [9, 15 and 16]. Unlike systems which represent knowledge in a relatively declarative and static way, the rule-based approach represents knowledge in terms of rules that can be used to classify members in different situations. In general, a good rule-based classification approach should maximize the distance among the members from different groups, and minimize the distance among members inside the same group.

As stated at the beginning of this section that shots filmed in the same scene should mostly share similar visual contents, here we appointed the extracted shot similarity as the main feature using the following rules:

(i) The first shot s_1 is always regarded as the beginning of the first scene.

(ii) For each following shot $s_i (i = 2, 3, \dots, N_i)$, the Shot similarity $ShotSim(S_1, S_i)$ between s_1 and s_i is calculated. The decision for whether s_i is the ending boundary of this scene is based on the obtained $ShotSim(S_1, S_i)$. If the $ShotSim(S_1, S_i)$ is above threshold, T_2 , which means s_1 and s_i are similar to each other, and then s_i will be grouped into the same scene as s_1 . Otherwise, if the $ShotSim(S_1, S_i)$ is below T_2 , s_i will be regarded as a scene

boundary, and stored in the ending-boundary set for further use. After looping through all the remaining shots, a temporal set of scene boundary is built using the given rules. For instance, the first detected temporal scene can be represented as $Scene(b, e)$, where b and e are the beginning and ending of this scene.

(iii) Then, the next shot s_2 is regarded as the temporal beginning boundary of the next scene to find the next temporal ending scene boundary using step (ii) above. Thus, given a video with n detected shots, $n-1$ temporal boundary sets will be built up after looping through all the shots.

(iv) Given groups of the temporal scenes estimated in previous steps, such as $Scene(b_1, e_1)$, $Scene(b_2, e_2)$, $Scene(b_3, e_3)$ and $Scene(b_n, e_n)$ the next step is to classify them into a more accurate scene group.

In order to improve the accuracy of the extracted scene boundary, we check the boundaries of each temporal scene from the first shot to the last shot to merge different scenes. Where for two neighboring scenes, $Scene(b_i, e_i)$, $Scene(b_{i+1}, e_{i+1})$, if the intersection of two range spaces $[b_i, e_i]$ and $[b_{i+1}, e_{i+1}]$ are not empty, then these two scenes are merged together and the beginning and ending boundaries will be updated accordingly, i.e. if $e_i < e_{i+1}$, the new scene boundary will be $Scene(b_i, e_{i+1})$, otherwise the boundary will be $Scene(b_i, e_i)$.

And then, loop through the remaining temporal scenes boundary to merge the similar scenes together. If the result of the logical operator AND for two scenes boundary sets are empty, these two scenes cannot be merged together and loop into the next scene. The above process will go through all detected shots until all the candidate boundaries are determined.

In addition, we also introduced a rule to merge short shots into previous ones for efficiency. For shots which only last for 5 to 30 frames, they are normally generated and edited by film director to bring audience some special feeling. For instance, the fast changing of the video shot can bring tension to the audience and it can be easily found at fighting and chasing scenes. Considering these shots may often have very sharp camera motion or fast object motion, the contents in every frame of one shot are changed in very high speed, which caused a very low shot similarity between this kind of shot and other shots, which leads to increase the false detection of scene boundary. In order to improve the precision and save the processing time, all the shots with frames less than 30 will be united into the previous shot.

Table 1 illustrates typical results from one clip of movie, which includes 21 shots from #13 to #33, in which candidate scene boundaries are found by using the rules above. Taking each shot as one potential starting boundary of one scene, its

corresponding ending boundary is given in the third column. Then, it is interesting to see that these ending boundaries do not restrain at the same shot, such like there are seven temporary ending boundaries at shot #35, and eight at #36, and only one at #41. To find out the reasons for such a boundary distribution, we reviewed that part shot by shot and concluded that it is due to special editing effects in movies to make false detection in the results.

As we know, various filming grammars may be applied by director to describe certain feeling of one scene, conterminous shots with content change in foreground or background alternatively, like in conversation and chasing scenes, or a shot filmed in different view point of the scene is interpolated to describe scene site or making certain feeling, which makes the visual content change repeatedly or sharply causing the similar change happens in the HSV field and scene boundaries are false detected.

In this situation, given an interpolated shot in one scene, the visual similarity between most of other shots and it is small enough to make it be regarded as the boundary of the scene, and make the real boundary be miss-detected.

Given the consideration above, we proposed a scheme to remove the false boundary candidates and find reliable or accurate boundary of each scene by using the proposed likelihood function in (7). This likelihood, L , which is designed to take into account the visual similarity and temporal frame distance $Temp_Dis$ between shot i and j , which can be formulated as:

$$L = \exp\left\{\frac{-M * |f_i - f_j|}{Total} * Vis_Sim(i, j)\right\} \quad (7)$$

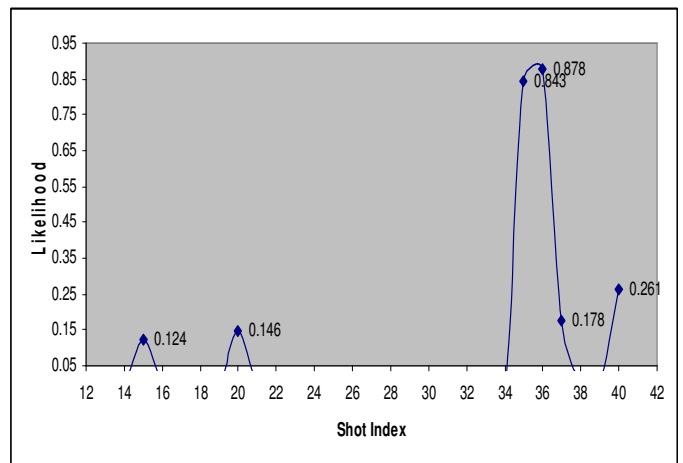
where M is the number of times the same shot be detected as the temporary scene boundary. f_i and f_j is the number of frames in shot i and shot j , respectively. Obviously, the likelihood L should be inversely proportional to the $Temp_Dis$. The constant $Total$ is the total number of frames in a video.

Given the rules (i-iv), the shots from #14 to #40 of 'Brave Heart' will be merged together and the scene boundary will be defined as $scene(14,40)$, however, the shot #40 is a false detection. Given the extracted the probability table from shot #14 to shot #40 illustrated in Table 2, it shows that the probability of shot #40 is only 0.261, which is lower than the value obtained at shot #36, which means shot # 40 should be a false detection, and the ending shot boundary will be determined as shot #36, which is also the same as the ground truth.

Table 1. Typical results of scene detection from shot #14 to shot #40 in 'Cave Ventura 2'

Temporal Scene	Starting Boundary at shot #	Ending Boundary at shot #
S1	14	15
S2	15	36
S3	16	35
S4	17	35
S5	18	20
S6	19	36
S7	20	40
S8	21	35
S9	2223	36
S10	24	35
S11	25	35
S12	26	36
S13	27	35
S14	28	36
S15	29	35
S16	30	36
S17	31	36
S18	32	37
S19	33	36

Table 2. Typical results of scene detection from shot #14 to shot #40 in 'Brave Heart'



3 Experimental Results

We evaluate the proposed scene boundary finding algorithm on three Hollywood movies: "Ace Ventura 2", "Cave", and "Brave Heart", one sample of the detected key-frame of each movie is shown in Figure 2. Each clip using for evaluation is

taken from part of the film lasts 20-40 minutes. The data set includes a variety of film genres, such like Epics, Comedy and Horror, which has various filming style from featured films to show the performance of the algorithm. The results show that our algorithm is robust for different film genre. For each video, the ground truth (*GT*) of the scene boundary is identified by ten human observers from our research group.



Figure 2. Three sample frames from the movies in our experiments. (a) Brave Heart, (b) Cave, (c) Ice Age 2 and their frame sizes are 720×416 , 640×272 , and 720×272 , respectively.

Table 3 summarizes the data set and the estimated results from our method with regard to the ground truth and false detected scenes. If the frame number of the detected scene boundary is the same as the ground truth table, it's a correct detection of scene boundary, and otherwise it's a incorrect detection. The recall and precision values are also provided to see the efficiency due to its good behaviour of data-retrieval assessment. The recall is defined as the ratio of the number of correctly identified scene boundaries to the total number of scenes in the ground truth, and precision is the ratio of the number of correctly identified scenes to the number of scenes detected is correct, where the higher these ratios are, the better the performance it is. The recall and precision is defined as following and more details can be found at [11]:

$$Recall = N_c / (N_c + N_m) \quad (8)$$

$$Precision = N_c / (N_c + N_f) \quad (9)$$

where N_c is the number of correctly detected scenes, N_m is the number of missing detected scenes, and N_f is the number of false detected scenes.

To justify the performance of our proposed method, we compare it with other existing methods using average Recall and Precision rate, details of the results can be found at Table 4. In comparison with those two others, our approach yield better performance in both precision and recall rate which has proved the effectiveness of the proposed rule-based methodology.

4 Conclusion

In this paper, we proposed a rule-based video segmentation method to determine scene boundaries. In our approach, both low-level features and high-level concepts rules are utilized to structure the video into a hierarchical structure, which is key-frames, shots, and scenes. In addition, the likelihood function is introduced to improve the accuracy of scene boundaries. This has inevitably suggested that these rules and the

corresponding likelihood function have provided practical ways in accurately determining boundary of video scenes. We have applied the proposed method on different film genres and found that our method has a better outperform comparing with several others in terms of average precision and recall rate in finding the scene boundaries.

Table 3 Details of the data set and the results of the algorithms with the ground truth

	Names of sequences		
	Ace Age 2 (AV)	Cave (CA)	Brave Heart (BH)
Duration (min)	36	20	26
Frames	53166	25949	47073
Shots	274	251	279
Experiment Results with the Ground Truth			
Scenes Detected	16	17	12
N_c	12	13	12
N_f	4	4	0
N_m	3	2	4
Scene Boundary in Ground Truth	15	15	16
Precision	75%	76.5%	100%
Recall	80%	86.7%	75%

Table 4. Comparison of average Recall and Precision rate with other algorithms

	Average Recall Rate	Average Precision Rate
Method in [5]	80%	63%
Method in [8]	74%	74%
Proposed method	80.56%	83.83%

Acknowledgements

The research leading to this paper was supported by European Commission under contracts FP6-027026 (K-Space) and FP6-027122 (Salero).

References

- [1] Cotsaces. C; Nikolaidis. N, Pitas. I; 'Video Shot Detection and Condensed Representation: A Review'; Signal Processing Magazine, IEEE, Vol. 23, No. 2. (2006), pp. 28-37
- [2] Yusoff Y, Christmas WJ, Kittler JV (2000) Video Shot Cut Detection Using adaptive thresholding, Proceedings

- of the 11th British Machine Video Conference, Bristol, UK.
- [3] Grana C; Cucchiara R; 'Linear Transition Detection as a unified Shot Detection Approach'; IEEE Transactions on Circuits and Systems for Video Technology : Accepted for future publication, Issue 99, 2006 Page(s):1 – 1
- [4] Otsuka, I.; Nakane, K.; Divakaran, A.; Hatanaka, K.; Ogawa, M; 'A Highlight Scene Detection and Video Summarization System Using Audio Feature for a Personal Video Recorder'; IEEE Transactions on Consumer Electronics, Vol. 51, Issue 1, Feb. 2005 Page(s):112 – 116.
- [5] Rasheed, Z.; Shah, M.;' Detection and Representation of Scenes in Videos'; IEEE Transactions on Multimedia, Volume 7, Issue 6, Dec. 2005 Page(s):1097 – 1105
- [6] Hui F, Jianmin J, Yue F: A Fuzzy Logic Approach for Detection of Video Shot Boundaries. Pattern Recognition 39(11): 2092-2100 (2006).
- [7] [7]. Chong-Wah N, Yu-Fei M, Hong-Jiang Z, 'Video Summarization and Scene Detection by Graph Modelling' IEEE trans on circuits and systems for video technology (vol. 15. no 2) Feb 2005
- [8] C-W. Ngo, T-C Pong and H. J Zhang; 'Motion-based Video Representation for Scene Change Detection', International Journal of Computer Vision 50(2): 127-142 (2002)
- [9] [9]. A. dorado, J. Calic, and E. Izquierdo, 'A Rule-Based Video Annotation System', IEEE trans on Circuits and systems for video technology, vol. 14, no.5, (622-633)May 2004.
- [10]R. S. Jadon, S. Chaudhury, K. K. Biswas, 'A Fuzzy Theoretic Approach for Video Segmentation Using Syntactic Features', Pattern Recognition Letter. 22 (2001) 1359-1369.
- [11]S. Porter, M. Mirmehdi, B. Thosmas, 'Temporal Video Segmentation and Classification of Edit Effects', Image Vision Computer. 21 (2003) 1098-1106
- [12]H. Fang, J. Jiang, 'Predictive-based cross line for fast motion estimation in MPEG-4 videos', Proceedings of IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology, USA, 2004, PP. 175-183
- [13]R. Jain, R. Kasturi, B. G. Schunck, 'Machine Vision', McGraw-Hill, New York.
- [14]Dubuisson. M. P, Jain. A K., 1994. A modified Hausdorff distance for object matching. In: Proc. 12th Internat. Conf. on Pattern Recognition, Jerusalem, Israel, Oct. 1994, pp. 566–568.
- [15]N. B. Karayiannis, A. Mukherjee, J. R. Glover, P. Y. Ktonas, J. D. Frost, Jr. Richard, A. Hrachovy, E. M. Mizrahi, 'Detection of pseudo sinusoidal Epileptic Seizure Segments in the Neonatal EEG by Cascading a Rule-Based Algorithm With a Neural Network', IEEE Transactions on BIOMEDICAL ENGINEERING, Vol. 53, no. 4 APRIL 2006.
- [16]Li, J, 'Robust Rule-Based Prediction', IEEE Transactions on Knowledge and Data Engineering, Volume 18, Issue 8, Aug. 2006 Page(s):1043 - 1054