ISSN 1746-7233, England, UK
World Journal of Modelling and Simulation
Vol. 1, No. 2, 2005, *pp*. 111-116

Video scenes clustering based on representative shots

Jun Ye [1], Jian-liang Li [2 +], C. M. Mak [3]

[1] Dept. Applied Math., Nanjing University of Posts & Telecommunications, Nanjing 210003, P.R.China
[2] School of Science, Nanjing University of Science &Technology, Nanjing210003, P.R.China
[3] Dept. of Building Services Engineering, The Hong Kong Ploytechnic University, Hong Kong, P.R.China

(*Received June 21 2005, accepted August 6 2005*)

Abstract. Clustering of video data is an important issue in video abstraction, browsing and retrieval. Movie is a kind of complex video with rich content. As for the movie, however, clustering is more complicated than other types of videos like surveillance, sport games, and documentaries. In this paper, we propose a novel shot clustering algorithm combing the editing feature of the movie and the criteria of choosing representative shots. The simulated experiment results demonstrate the effectiveness of our method for the dialogue-dominated movie.

Keywords: shot clustering, representative shot, editing feature, color and edge retrieval

1. Introduction

Rapid advances in multimedia processing, computing power, high-speed internet working, and the World-Wide Web have made digital videos an important part of many emerging applications such as distance learning, digital library libraries, and electronic commerce. Searching for a desired video segment from a large collection of videos becomes increasingly more difficult as more digital videos are easily created. A well-known search approach matching user-specified keywords with titles, subjects, or short text descriptions is not effective because these descriptions are too coarse to capture rich semantics inherent in most videos. As a result, a long list of search results is expected. Users pinpoint their desired video segment by watching each video from the beginning or skimming through the video using fast-forward and fast-reverse operations. Content-based video browsing and retrieval is an alternative that lets users browse and retrieve desired video segments in a non-sequential fashion [1]. Consequently, the content-based access and retrieval become a proper solution.

Video content can be grouped into two levels: low-level visual features and high-level semantic content. Low-level visual content is characterized by visual features such as color, shapes, texture, etc; On the other hand, semantic content contains high-level concepts, such as objects and events. Because it is difficult to map the low-level features to semantic content, currently, most video retrieval systems rely on low-level features and video annotations [2].

Clustering is a natural solution to abbreviate and organize the content of a video. Many clustering algorithms have been proposed. Chong-Wah Ngo et. al [3] utilized the tensor histogram for motion feature extraction, and used the k-mean algorithm to sports video. Alan Hanjalic et.al [4] presented a newly developed strategy for automatically segmenting movies into logical story units. They understood the logical story unit as an approximation of a movie episode, which was a high-level temporal movie segment, characterized either by a single event (dialog, action scene, etc.) or by several events taking place in parallel. A method of statistical approach using Hidden Markov model to classify movie scenes proposed by Yuan-Kai Wang and Chih-Yao Chang [5]. They classified two important kinds of movie scenes, dialogue and fighting scenes.

In this paper, based on Lu Hai-Bin's work [6] on the criteria of choosing representative shots and Ying Li's work [7] on the algorithm of WBS (Window-based Sweep Algorithm), we propose a novel shot

+ Corresponding author.
Email address: ljl6006@hotmail.com.

clustering algorithm combing the editing feature of the movie. From the simulation results, we demonstrate that our method is prior to the WBS algorithm in time consuming, the precision and the recall, respectively.

2. Video Scenes Clustering Based on Representative Shots

2.1. WBS Algorithm

It was a shot clustering algorithm which grouped shots into shot groups. Given shot i, that algorithm found all shots that were visually similar to i, and pushed them into the same group. Actually, the WBS algorithm is an ordinal circular clustering method to the shots that have been segmented.

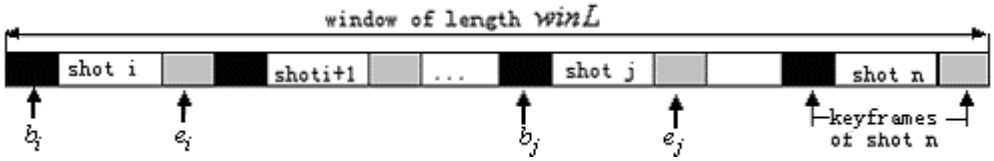

Figure 1 Shots contained in a window of length $winL$.

They computed the similarity between shots i and j by

$$Dist_{i,j} = \frac{1}{4}\left(w_1 \times dist\left(b_i, b_j\right) + w_2 \times dist\left(b_i, e_j\right) + w_3 \times dist\left(e_i, b_j\right) + w_4 \times dist\left(e_i, e_j\right)\right) \quad (1)$$

Where $dist\left(b_i, b_j\right)$ could be either the Euclidean distance or the histogram intersection between b_i and b_j's color histograms. b_i, e_i represent the first frame and the last frame of the shot i as shown in figure 1. w_1, w_2, w_3, and w_4 are four weighing coefficients. Also, since an event practically occurs within a certain temporal locality, they naturally restrict the search range to a window of length $winL$ [7] as shown in figure 1.

Basically they will run that algorithm for every shot. To a shot sequence of n shots, the comparative number of WBS algorithm is $\frac{n\left(n+1\right)}{2}$. The complexity of computing is very high.

2.2. Clustering Based on Representative Shots

In order to lessen the complexity of computing, we propose a novel clustering algorithm based on the criteria of choosing representative shots (CBRS). Since keyframes can be seen as the shot representatives in most cases, we can also utilize the representative shots to represent the scene. So how to choose the representative shots become the key problem. In our paper, we use the criteria of representative shots proposed by Lu Hai-Bin. They defined the criteria of choosing representative shots according to the custom of video editing. Two kinds of shots are representational. One type are the recurrent shots, and the other are the shots that last for a long time [6]. Shot groups are generated using the proposed CBRS algorithm as described below.

CBRS Algorithm: Given the representative shot i, this algorithm finds all shots that are visually similar to i in a neighborhood, and cluster them into the same shot group. The restrictive range is the neighborhood as shown in fig. 2.

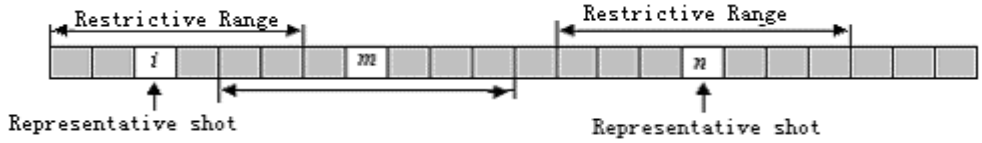

Figure 2 Shot Clustering based on the Representative Shots

We cluster the shot in a neighborhood since a scene practically occurs within a certain range of time. An actual example in figure 3 shows the necessity of the restrictive range.

In fig.3, it shows two shots' key-frame, respectively. Fig. 3(a) shows a dialogue shot with two people. Fig. 3(b) shows that two persons are talking to another one. With the popularly similar formula, the similarity between shot 7 and shot 38 is very high. So they will be pushed into the same shot group. Actually they describe different scenes.

(a) key-frame of the 7# (b) key-frame of the 38#

Figure 3 The Key-Frame of Shot

We define the restrictive range by

$$dist(i,j) = |i-j| \tag{2}$$

Where i represent the representative shot, and the shot j represent other shots.

If $dist(i,j) \le C$, then we think that the shot j is a candidate clustering shot. Otherwise, we overleap that shot. Here, C is a constant. Constrained with the condition (2), we will then put the shot 7 and shot 38 into different groups.

In order to compute the similarity between two shots, we also utilize a shot's two frames –the first and the last as its key frames. We compute the similarity between representative shot i and the candidate clustering shot j by

$$Sim(s_i, s_j) = w_1 \times S(b_i, b_j) + w_2 \times S(b_i, e_j) + w_3 \times S(e_i, b_j) + w_4 \times S(e_i, e_j) \tag{3}$$

where $S(b_i, b_j)$ combined two kinds of similarity to synthetically predict the similarity between shot i and shot j's first frame.

The $S(Q,I)$ is defined as

$$S(Q,I) = \alpha \cdot S_1(Q,I) + (1-\alpha) \cdot S_2(Q,I) \tag{4}$$

where α is the weighing coefficient.

The $S_1(Q,I)$ is the histogram intersection between frame Q and frame I's color histograms, and its normalized formula is defined as

$$S_1(Q,I) = \sum_{i=0}^{k-1} \min\left(h_Q(i), h_I(i)\right) / \min\left(\sum_{i=0}^{k-1} h_Q(i), \sum_{i=0}^{k-1} h_I(i)\right) \tag{5}$$

where $h_Q(i), h_I(i)$ signify the color histograms of the frame Q and I.

Texture in images has been recognized as an important aspect of human visual perception. To compare the edge features between two images, G. Sheikholesl et. al [9] used the two-level wavelet transform as shown in figure 4.

Different sub-bands generated by wavelet transform have information about horizontal, vertical, and diagonal edges, which helps in extracting features related to directionality of images[9]. They calculated the mean and variance of wavelet coefficients to represent the contrast of the image. They also count number of edge pixels in horizontal, vertical, and diagonal directions to have an approximation of directionality of the image. So the feature vector for each of those sub-bands is < mean, variance, number of edge pixels >. The total features of those are $18(6\times3)$.

The $S_2(Q,I)$ is the textural similarity between frame Q and I using the above wavelet transform, and it is defined as

$$S_2(Q,I) = \sum_{i=1}^{18} s_i(Q,I) \cdot w_i = \sum_{i=1}^{18} \frac{\min\left(f_i^Q, f_i^I\right)}{\max\left(f_i^Q, f_i^I\right)} \cdot w_i, \sum_{i=1}^{18} w_i = 1, w_i \ge 0, i = 1,...,18. \tag{6}$$

Where $(w_1, w_2, ..., w_{18})$ is a weight vector, and the w_i is the weight that is assigned to the feature f_i.

And $f_i^Q, 1 \le i \le 18$ represent the wavelet feature of the frame Q.

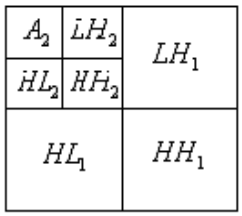

Figure 4 The seven sub-bands generated after applying two levels of wavelet transform

2.3. Choosing representative shots based on the editing feature

According to the criteria of the representative shots, we first choose the shots that last for a long time. We statistic to all shots that have been detected in the video and get the average shot length $\overline{L}$.

If shot i's length $L_i > c\overline{L}$, then we will think that the shot is the representative shot. Otherwise, we will check the next shot.

After choosing the shots that go on long time, we group those representative shots into shot groups using the CBRS algorithm. In order to choose the recurrent representative shot, we use the montage effect in a film and the shot groups that have been clustered. In video, a single shot presents an image of time. In motion pictures, a story is told by the presentation of the images of time: the sequence of such presentation is called the montage [8]. The simplistic model which can represent the montage effect is "ABABABA", where A and B represent two people in a dialogue scene, respectively. Compared to the other shot, we can see that the shot in that model is more frequent. So we choose the shot in the montage effect model as the recurrent shot.

The algorithm of clustering based on representative shots is described as

Algorithm:

Input: Shot Sequence $s_1, s_2, ..., s_n$, where n is the initial number of clustering shot group .

Output: Shot Groups $GS = \{rgs_1, rgs_2, rgs_3, ..., rgs_l\}, 1 < l < n$.

Step 1 Compute the average shot length $\overline{L}$. If the shot i's length $L_i > c\overline{L}$, the shot i is chosen as the representative shot. Where c is a constant, whose choosing is based on the video type.

Step 2 We group the similar shots into shot groups using the CBRS algorithm according to the formula (3). If $S(s_i, s_j) > T, 1 \le j \ne i \le n$, then the shot i and j are clustered into a same group, where T is the similarity threshold. The obtained group is represented by

$$rgs = \{gs_i, gs_j, gs_k, ..., gs_m\}, 1 \le i, j, k, m \le n$$

Step 3 Checking the list number in the clustered shot groups. If they are complete or partial arithmetic progression, then we get next shot of the list of arithmetic mean as our recurrent representative shot; If they are not, go to Step 4; For example, if shot i's shot group contains shots i, $i+2$, $i+4$, then we will choose the shot $i+3$ as the recurrent shot. Then we cluster recurrent shots into the shot groups using the CBRS algorithm according to the formula (3).

Step 4 For the rest shots in the video, we group the similarity shots into other shots groups using the CBRS algorithm.

3. Experiment Results

To show the effectiveness of the proposed algorithm, we contact experiment on the film clip named "Meet the parent", it is a typical dialogue drama film.

The clustering performance is evaluated in terms of recall and precision where

$$precision = \frac{number\ of\ right\ shot\ groups}{total\ number\ of\ shot\ groups\ grouped\ by\ given\ method}$$

$$recall = \frac{number\ of\ right\ shot\ groups}{total\ number\ of\ shot\ groups\ grouped\ manually}$$

Recall measures the ability to present all relevant items, while precision measures the ability to present only relevant items. Recall and precision are in the interval of [0,1].

First, we segment the film clip into shots using the algorithm for shot boundary detection that integrates the spatial and color features of the frames proposed by Cheng Yong et.al [10]. Their method is not sensitive to brightness change and quick motion. And it can choose the threshold automatically. There are 56 shots extracted in the experimental film clip using the above method.

We group the similar shots into shot groups using the method of the representative shots.

The restrict range is set to be $C = 12$, which ensures the validity of the clustering groups. When T is 0.70 (in Step 2), c is 2(in Step 1), α is 0.70, and $w_1 = 0.15$, $w_2 = 0.10$, $w_3 = 0.60$, $w_4 = 0.15$ in the formula (3) in algorithm, the results show more reasonable and get better results. All these parameters are experiential values. One thing worth mentioning is, if shot j does continue shot i's content, $S(e_i, b_j)$ should have the biggest similarity. So we set the weighing coefficient w_3 greater than other weighing coefficient. Fig. 5 shows the first and the last frames of each shot in a clustered shot group. It reflects the shot group including the shot of 47#, 49#, 51#, 53#, 55#. And we could find the recurrent shot according to that shot group.

Figure 5 The Shot Group

Finally, all the shots are clustered into different shot groups according to the content similarity. The data in our experiments are shown in Table 1.

From the data, we can see that the precision and recall of our method are better than the WBS method in shot clustering. From the point of the complexity of clustering, the CBRS algorithm is also better than the WBS algorithm. By the way, the time consuming in our experiment outgo the WBS algorithm in the same operating environment. Our experiment platform is MATLAB 6.0, so relative to the C++ platform, the time-consuming is long.

Table 1 Experimental Results for Shot Clustering

	CBRS Method	WBS Method	Manual Method
Total of Shot Groups	21	23	19
Wrong Shot Groups	5	8	0
Right Shot Groups	16	15	19
Precision	76.2%	65.2%	100%
Recall	84.2%	78.9%	100%
Time Consuming	15.47 min	18.92 min	

4. Conclusion

we propose a new algorithm based on the editing feature and the criteria of choosing representative shots, which lessens much computation efforts and time in the shot clustering. Also the precision and the recall are

both improved relative to the WBS algorithm in the same experiment platform. In the future, other editing features could be utilized in the clustering algorithm to improve the accuracy.

5. Reference

[1] Wallapak Tavanapong, and Junyu Zhou, *Shot clustering techniques for story browsing,* IEEE Trans. Multimedia, 6(2004)4, pp. 517-527.

[2] Wen-Gang Cheng, De Xu, *Content-based video retrieval using the shot cluster tree*, In: Proc. of the second IEEE Int. Conf. on Machine Learning and Cybernetics, Xi'an, China, pp. 2901-2906, 2003.

[3] Chong-Wah, Ting-Chuen Pong, Hong-Jiang Zhang, *On Clustering and retrieval of video shots*, NEC Research Institute, 2001.

[4] Alan Hanjalic, Reginald L., Lagendijk, Jan Biemond, *Automatically segmenting movie into logical story units*, Faculty of Information Technology and Systems Information and Communication Theory Group. NEC Research Institute, 1999.

[5] Yuan-Kai Wang, and Chih-Yao Chang, *Movie scene classification using hidden markov model*, Proceedings International Conference on Computer Vision, Graphics and Image Processing, 8(2003), pp. 196-201.

[6] Hai-Bin Lu, Yu-Jin Zhang, and Wei-Ping Yang, *Shot and episode based nonlinear organization of video*, Chinese Journal of Computers, 5(2000).

[7] Ying Li, Shrikanth Narayanan, and C.-C. Jay Kuo, Fellow, *Content-based movie analysis and indexing based on audio visual cues*, IEEE Trans. Circuits and Systems for Video Technology, 14(2004)8.

[8] Minerva M. Yeung, and Boon-Lock Yeo, *Video content characterization and compaction for digital library applications*, in Proc. SPIE, 3022(1997), pp. 45-58.

[9] Gholamhosein Sheikholesl, Aidong Zhang ,and Ling Bian, *A multi-resolution content-based retrieval approach for geographic images* , Geo Information, 3(1999)2, pp. 109-139.

[10] Yong Cheng, and De Xu, *A method for shot boundary detection using adaptive threshold*, Chinese Journal of Electronic, 3(2004), pp.508-511.