The Structured Analytical Technology of Video Information Management

Jun Zhou Zefu Lin Yaoqi Wang

School of Management Science and Engineering, Central University of Finance and Economics, Beijing 100081, P. R. China

Abstract

When the information management system is carried into execution, the unstructured video data always cannot be organized properly. Traditional analytical method by handling large number of video shots cannot convey meaningful semantics. We divided the video into appropriate shots with the improved twin-comparison method to gain effective browsing and retrieval purpose. So the convenient technique of generating video scene structure is set up, through taking a method of sliding shots window, shots are grouped into clusters to improve the efficiency and effectiveness. Finally a demonstration analysis shows it is a useful way.

Keywords: Analysis, Video data, Information management, Semantic unit, Structure

1. Introduction

Recently, digital project and information highways construction make great progress; more and more video data are being captured, produced and stored in multimedia system. However, without appropriate techniques that can make the video content more accessible. So it is necessary to organize the unstructured video information properly.

Video information organization is based on individual video shots in a general way, which is used as the smallest logical unit. A shot is defined as a sequence of frames recorded contiguously and representing a continuous action in time or space. While existing shot-based video analysis approaches provide users with better access to the video than the raw data stream does, they are still not sufficient for meaningful video browsing and retrieval. Firstly, it is common that a movie about one hour contains several thousands shots. In addition to the large number, human understanding of video content is far more concentrated on meaningful and high-level semantic video units rather than on single shots. A video scene is defined as a sequence of semantically correlated shots near in time or location. So the construction of scene is thus of fundamental importance to many video applications, such as video abstraction, indexing and browsing.

Scene information construction can be classified in two entries: model-based and general propose. In the first approach, a prior model of a particular application or domain is first constructed. Swangberg's theoretical framework has been used in news video parsing and TV soccer program parsing. But it requires good domain knowledge and the domain model must be constructed for each application. As for the other, a method had been set up based on the principle of STG (Scene Transition Graph) [1], [2]. In their researches, the emphasis was put on the joint use of features extracted from audio, video and textual information; the same way relied on statistical techniques like shot boundary detection [3], [4]. Time-constraint clustering method and its improved algorithm were used in to group the shots and construct the video scene information. However, all of above method is complicated and time-consuming. In this paper a new scheme of generate video scene structure automatically and quickly in two steps: shots clustering and shots correlations analysis.

2. The structured analytical technology of video information management

At first, a scene as a sequence of semantically correlated shots near in time or location is defined. The “shot cluster” is a basic unit of scene and we define the “shot cluster” as a group of shots similar in content and near in time or location. If we examine some video documents, we can find a scene may be either a shot cluster or some shot clusters interacted. A scene consisting of only one-shot cluster means those shots in the cluster is the same thing or may be recorded at same place sharing the same background. Such as scenery in the forest, if several different shot clusters make up of a scene, the reason may be as follows: As video shots of different shot clusters appear in turns, it is reasonable that their topics are closely related or even the same. The most common example is a dialog scene between two people. There are shots taken at the one of them, the other, and both of them. Then shot clusters containing those shots always intersect with each other throughout the whole talking.

2.1. Video shots and key frame

Boreczky and Rowe compared several shot boundary detection techniques on real video sequences and found that twin-comparison method is a simple algorithm that works very well [5]. The twin-comparison method can detects both abrupt and gradual transitions at the same time, but it has a problem: In fact, there are some gradual transitions during which the consecutive frame difference may falls below the lower threshold. However the original method will miss this kind of transition. We solved this problem by setting a tolerance value that allows a certain number of consecutive frames with low difference values before rejecting the transition candidate. Histogram-based two-comparison method is used to detect the shot boundary. And then, each shot can be represented by a number of key frames. In our test, we regard the middle frame of each shot as the key frame just to reduce the computation efforts. Although we believe the performance of algorithm can be improved using more sophisticated key frame extraction methods.

2.2. Shot clustering

In this step, we will collect the similar shots into same shot cluster. Clustering is unsupervised learning classification method. There are mainly two types of clustering: partition clustering methods and hierarchical clustering methods. For we have single shots at hand, an agglomerative hierarchical clustering method is fit to do that.

There are too many shots in a longer video document. If we just use the simple hierarchical clustering method to cluster the data, it is evident that it will take too much computation efforts and time. Because the shots in a same scene are close to each other, we can reduce the number of shots to be compared. In Proc: IEEE Int. Conf. on Multimedia Comput. And Syss [6], the author proposed a time-constrained clustering approach to grouping shots, where the similarity between two shots is set to 0 if their time difference is greater than a predefined threshold. Rui gave a method of time-adaptive grouping approach, the similarity decreasing with the time difference becoming bigger [7]. Though simple and less computation efforts than before, the above two methods have many demerits. Besides comparing too many shot pairs, they still have another two drawbacks. First, after video editing, there are many shots whose length may vary from each other dramatically. A video edited using few long shots inspires quietness, while videos with many short shots emphasize dynamism and happiness. So the time threshold may be improper to handle video shots, and a threshold determined by the number of shots seems more reasonable. Second, in a certain video clip, the similarity between two shots should not change according to their time difference. For example, in a dialog scene, the first shot may be quite the same as the last one. The similarity between them may be lower than that of other two different shots in content just because the former two shots are far away from each other, which is unreasonable.

So we put up a shot clustering method using the sliding shot window (SSW). SSW is a window that each shot is to be compared with the shots in its window, which can help reduce the number of comparing times and solve the above two problems. If we assume the SSW's width is 2L (L is the number of shots), we just calculate the similarity between the current shot (we refer it as Current Shot, CurShot) with L shots (We refer it as Destination Shot, DestShot) or the clusters that the DestShot belong to if the DestShot have been group into a cluster, before and after it. When the CurShot changes into the next shot, the window moves consequently.

Color is one of the most cognizable and important elements of visual content, and is widely used because of its invariance with respect to image scaling, translation and rotation. So we use the color feature to represent the content of the key frame, while the latter can describe the content of the whole shot.

The similarity between shot i and shot j (Fi, Fj is their key frame respectively) is measured by:

$$Sim(i,j) = \sum_{k=1}^{N} min(HF_i(k), HF_j(k))$$

Where HFi and HFi are the normalized histogram for the two key frames and N is the number of bins used in the histogram. In other words, we use histogram intersection to describe the similarity. The color histogram is measured in HSV color space. We adopt the HSV color space because it is supposed to provide better correspondence with human visual perception of color similarities than for example RGB color space. According to the different color ranges and human perception of color, we quantify the three channels of HSV and a vector is used.

We call a group of some similar shots as a shot cluster (SC). We define the similarity between a shot i with a SC k as follows:

$$SimSSC(i,k) = \max_{j \in SC\ k}(Sim(i,j))$$

where j is a shot of SC k.

The algorithm of clustering method using SSW is described below:

Algorithm:（Hierarchical clustering of shots near in time or location using SSW）

Input: Shot Sequence: Shots={s 1,s 2,…,s M}

Output: Shot Clusters: ShotClusters={SC 1, SC 2,…, SC N} $(1 \le N \le M)$

Procedure:

Step 1: Initialization. Input Shot Sequence. Denote the current shot as the first shot in sequence. CurShot=s 1.

Step 2: If the CurShot is Null, quit; otherwise, go to the step3.

Step 3: Compute the similarity between the CurShot (or the SC the CurShot belongs to if the CurShot has been grouped into SC) and DestShot (or the SC the DestShot belongs to if the DestShot has been grouped into SC) in the SSW. If the similarity lower than a threshold T, merge them into a SC. Otherwise, form a SC with the CurShot only (If the CurShot has not been grouped before). Go to Step 4.

Step 4:The next shot in the sequence becomes the CurShot. Goto Step 2.

During the executing of the above procedure, in fact there is no need to compare the CurShot (or the SC the CurShot belongs to if the CurShot has been grouped into SC) with the DestShot (or the SC that the DestShot belongs to) before the CurShot. Because the similarity between any previous shot and CurShot has been achieved when the previous one served as the CurShot. So compare the CurShot (or the SC the CurShot belongs to if the CurShot has been grouped into SC) with L shots next to it only. The Figure 1 is an example (where the shot i-3 makes up of the SC j-1; SC j consists of two shots: shot i-2 and shot i-1; Shot i is the CurShot; L=3):

When the three shots before shot i served as the CurShot, they have been compared with the Shot i. So when the CurShot is shot i, we only compare it with next three shots.

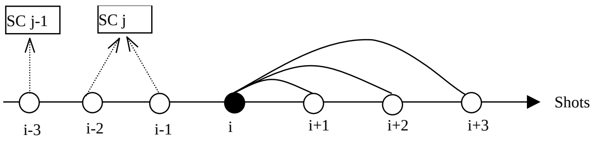

Figure 1: Shot clustering using SSW.

For its less comparing times and computation efforts, it is a simple and quick algorithm. After that, each shot in the shot sequence can be grouped into a certain SC.

2.3. Analysis of correlations between shot clusters

The shots in a SC are similar in content and near in time or location. So SC is the basic element of a scene. According to the development of video content, we divide the processing of video content into two types: Sequence Development and Interaction Development. The fist means something happens after other finished. In this case, a SC is a scene. The other one means two or several things take place simultaneously, but they have to be displayed sequentially. So video shot must focus on one, then another, or whole of them. In this case, several SCs make up of a scene and there are strong correlations between them

Now, we analysis the original shot sequence and substitute each shot ID with the corresponding SC ID. For example, the original shot sequence is 123456789, as Fig. 2.

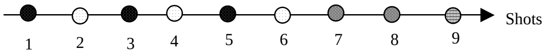

Figure 2: The original shot sequence.

Assuming shot 1,3,5 are grouped into SC A, shot 2,4,6 into SC B, shot 7,8 into SC C, and shot 9 into SC D. The sequence now becomes ABABABCCD. It is evident that A and B are developed in the second type: Interaction Development, while C and D in the first type: Sequence Development.

Now we define a function to measure the correlations between two scenes SC x and SC y:

$$Cor(x,y)=\begin{cases}\frac{1}{2}*(\frac{InMid(x,y)}{Count(x)-1}+\frac{InMid(y,x)}{Count(y)-1}) & x \neq y \\ 1 & x = y\end{cases}$$

where InMid(x,y) means how many times y appears between each two adjacent x; Count(x)-1 means the total number of two adjacent x.

As in the above example, we can get Cor(A,B)=1, while Cor(C,D)=0. Which means the SC A and SC B are in strong interaction, them two should be in a same scene consequently; The SC C and SC D are take place separately, and they should form two different scenes.

The correlations between all the SCs form a matrix. According to the constraint of SSW, each SC has its time property. So if the correlations between some SCs are greater than 0, those SCs make up of a scene. As a result, if an isolated shot appears between two similar shots, it will be included into the same scene as its neighbors, which is common and reasonable in video editing. The final scene structure can be generated after analysis of the correlations matrix.

3. Demonstration

The proposed algorithm is tested on four digital video clips, each of them from a certain video type. The Snow White and the Seven Dwarfs (SW) is a well-known movie; China (CN) is a documentary about Chinese history; World Cup Story of Baggio (BA) is a sports documentary from CCTV5; act 1 of Family Album U.S.A Ⅰ (FA) is an education video for learning English.

First, We segment the each video clip into shots using the improved two-comparison method. L1 distance (sum of absolute differences) is used to measured frame difference. We set the high threshold as 0.40 and low threshold as 0.15, which gets good performance for shot boundary detection in our experiments. And then, we group the similar shots into Shot Cluster using the method of SSW. It is evident that the types of video clips determine the window's size. A size big enough is used for all the types in our test. The window's size is set to L=12, which ensures the right shots should not be excluded from a certain scene. When T is 0.60 in algorithm (in Section 2.2), the results seem more reasonable and get better results. After the analysis of correlations between shot clusters, the last result shows in Table 1.

Video Clip	Frames	Shots	Shot Clusters	Scenes	Actual Scenes
SW	10089	69	23	13	10
CN	16320	45	38	29	27
BA	12006	109	37	24	18
FA	7813	49	11	4	4

Table 1: Experiment results by implementing method

The number of actual scenes is determined after watching the corresponding video clip. As we can see from the data, the proposed method shows better results.

4. Conclusions

The simple method of generating content video information is brought up and tested. An improved twin-comparison way is used for video segmentation into shot. We grouped the shots similar in content and near in time or location into shot clusters using a method of SSW and the shot clustering method is simple and quick for its lower computation efforts and less comparing times. A function of correlations between Shot Clusters is defined and measured to construct the final scene information. The method is demonstrated effectively.

References

[1] A.G. Hauptmann, M. A. Smith. Text, speech, and vision for video segmentation: The informedia project. *AAAI Fall Symposium, Computational Models for Integrating Language and Vision*, Boston, 1995.

[2] R. Lienhart et al. Scene determination based on video and audio features. *Technical report*, University of Mannheim, 1998.

[3] J.S. Boreczky, L. D. Wilcox, A hidden Markov model framework for video segmentation using audio and image features. I*Proc: Int. Conf. on Acoustics, Speech, and Signal, Seattle*, 1997.

[4] G. Iyengar, A. B. Lipman. Models for automatic classification of video sequences. *Proc: SPIE Storage and Retrieval for Image and Video Databases VI*, 1998.

[5] J.S. Boreczky, L.A. Rowe, Comparison of video shot boundary detection techniques. *Proc: SPIE, Storage and Retrieval for Still Image and Video` Databases IV*, San Jose, 1996.

[6] M. Yeung, B.L. Yeo, B. Liu, Extracting story units from long programs for video browsing and navigation. *Proc: IEEE Int. Conf. on Multimedia Comput,* and Syss, 1996.

[7] Y. Rui et al, Exploring video structure beyond the shots, *Proc: IEEE Int. Conf. on Multimedia Computing and Systems*, Texas, USA, 1998.