International Journal of Computer Sciences and Engineering Systems
Vol. 11 No. 1 (June, 2017)

Automatic Video Partition for High-Level Search

Song-Hao ZHU[1] and Yun-Cai LIU[2]

[1]Institute of Image Processing and Pattern Recognition, Shanghai Jiao tong UniversityShanghai 200240, China
E-mail: playtree@sjtu.edu.cn
[2]Institute of Image Processing and Pattern Recognition, Shanghai Jiao tong UniversityShanghai 200240, China
E-mail: whomeliu@sjtu.edu.cn

Abstract: *Browsing video scenes can help users to locate their desired video segments quickly and efficiently. Therefore, automatic segmentation of a long video archive into scenes is the crucial step toward content-based representation for indexing, browsing and retrieval purposes. In this paper, a novel scene segmentation scheme for various video types is proposed. Firstly, video shots are detected by using a coarse-to-fine algorithm. Secondly, the number of key frames within a shot is extracted by analyzing the dynamic content. Finally, spatio-temporal coherent shots are clustered into the same scene taking into account both the temporal constraint of video content and the visual similarity of shot activity. The proposed algorithm is conducted on various types of videos and the results demonstrate that the proposed method is a valid approach and encourage further research.*

Keywords: Content-based retrieval, video scene segmentation, two-dimension entropy, time-constraint

1. INTRODUCTION

Recently, the exponential growth of digital data in has led to a search problem and the manual searching process is low efficiency and time-consuming. Therefore, using which strategy to manage video archives efficiently and providing which index to retrieve quickly is becoming an open problem. Up to now, the most used approach is keywords, which fits well for text-driven database. However, for the visual information, browsing and retrieving method based on text is obviously not the effective solution. As a result, it is necessary to present a method to organize, index and retrieve video archives in view of semantic level.

According to the general perspective, the structure of video file is: frame, shot, scene and video. Frame is the lowest level in the hierarchical structure. A shot is the basic unit of a video, which consists of a series of adjacent frames with invariable background. In some cases, shots can achieve the browse and retrieval purpose, such as the case of less camera motion. For a typical produced video, however, the amount of shots could be large, it is hence necessary to provide a more concise and compact semantic segmentation to improve the performance. A scene reflects a certain topic or theme and using the scenes to browse and retrieve is obviously higher that based on shots. At the top level of the hierarchical structure is the video stream, which is composed of entire scenes.

1.1 Related Work

Recent years have seen an explosive increase of research on automatic video segmentation techniques. According to what discussed above, automatic video stream file segmentation includes three main steps. The first step is shot boundary detection. Much work has been reported in this area and highly accurate results have been obtained such as in [1]-[7]. The second step is key frame selection based on both the complexity of activity content and the duration of shot. In recent years, there are several contributing products include [8]-[13]. The last step is the scene segmentation where related shots detected are clustered into meaningful segments and the resulting scenes can provide a concise and compact index for the purposes of browsing and retrieving. How to define a scene based on human understanding is still an open topic and the segmentation of scene is the crucial step in the whole video stream file segmentation process. Recently a large amount of techniques have been reported to in this area involving [14]-[18], [24]-[25].

Generally speaking, video stream file can be classified into following types: news video, sport video, feature film, television video, home video, and so on, and much work have been reported about them. Jiang *et al.* in [19] used the features of audio and visual to complete the scene segmentation of news video and sport video. Hari *et al.* in [16] described a strategy of the segmentation of a film. Similar to [19], the audio and video data are first segmented into corresponding scenes separately. However, unlike the strategy of integration used in [19], the final scene boundaries are determined using a nearest neighbor algorithm. Boreczky *et al.* in [15] used hidden Markov models to search scene

boundaries. Features for segmentation include audio, visual and motorial. Hidden Markov models contain seven states, and the parameters trained include both seven transition probabilities and Gaussian distributions. Yeung *et al.* in [14] presented an approach for the segmentation of video into story units by applying a scene transition graph, STG. Based on the complete link technique for hieratical clustering, STG is then further split into several sub-graphs and each one presents a scene. Rasheed *et al.* in [17] proposed a graphical representation of feature films by constructing a shot similarity graph, SSG. The meaning of nodes and edges is same to that in [14]. Shot similarities are firstly computed by utilizing the information of audio and motion of the video. Then, the normalized cut method is used to split the SSG into smaller ones presenting story units. Tavanapong *et al.* in [18] exploited film making technique to cluster visually similar shots into the same scene. Based on the average value of all DC coefficients of each shot, the clustering of shots is done by forward comparison, backward comparison and temporal limitations.

However, several of methods of new video are not fit for feature film. On the other hand, other methods do not fully taken into account the characteristic of film editing, such as the linking of shots and scene is determined what coefficients. Zhai *et al.* in [24] used the Markov Chain Monte Carlo technique to determine the boundaries between video scenes. The temporal scene boundaries are detected by maximizing the posterior probability of the model parameters However, the Markov Chain Monte Carlo technique is too time-consuming.

1.2 Proposed Approach

Based on our daily experience and film editing principle [20], the relationship between successive scenes of feature film will comply with human intuitive understanding and film editing rules. For example, when the content of last shot describes a tumultuous event, then the next one will often start with a relative quiet scene. Such a transition can help to relax our highly intense mood and be ready to enjoy another wonderful scene. And the opposite often occurs in feature film too. The transition, this time helps to lead us into next highlight. Based on above careful observation, it is easy to notice the content of successive segments is often difference. This phenomenon inspires us to define an appropriate form to perform the detection of shot boundary. In this paper, we present two-dimension entropy model to perform the segmentation of shots and further group visually similar shots into the same scene. To avoid the under-segmentation of scene, a temporal constraint analysis is incorporated into the process.

The remainder of the paper is organized as follows. Section 2 introduces the two-dimension entropy algorithm and proposes the scene boundary detection method. Section 3 presents the segmentation of scene boundary. Section 4 describes experimental results. Section 5 concludes and discusses the proposed framework.

2. BACKGROUND AND PROBLEM FORMULATION

In this section, we will describe the concept of entropy, and formulation of our proposed two-dimension entropy.

2.1. Conception of Entropy

According to [20], one meaning of entropy is to measure the information and uncertainty of a random variable. Specifically, from a mathematical descriptive perspective, if a single random variable is depicted by x, $s(x)$ is the set of values that x can take, and $p(x)$ is the probability function of x, $H(x)$, the entropy of x, is then defined as shown in following equation:

$$H(x) = -\sum_{x \in S(x)} p(x) * \ln[p(x)] \tag{1}$$

Similarly, the mathematical formulation for entropy of a multivariate vector $\tilde{x} = \{x_1, x_2, ..., x_n\}$ can be computed:

$$Ent(x) = -\sum_{x \in S(x_1)} ... \sum_{x \in S(x_n)} ... p(\tilde{x}) * \log[p(x)] \tag{2}$$

where $p(x) = p(x_1, x_2, ..., x_n)$ is the multivariate probability distribution.

2.2. Formulation of Two-Dimension Entropy

Before going to the detail description of the formulation of two-dimension entropy model, there are some notations which should be introduced in advance. Given one pixel in an image t, the notation (p, q) is its typical descriptive character where p is its gray information and q is corresponding average gray information in its eight-connectedness region respectively. The gray level that p and q can take is $[0, 1, 2\ldots L\text{-}1]$. Generally speaking, L is set to be 256 for gray pixel. Let $f_{pq}^{\ t} = f(p, q)/r$ denote the joint character of notation (p, q), where $f(p, q)$ shows the frequency of (p, q) occurrence in image t, r is the size of image t. The formulation of two-dimension entropy model of (p, q) in image t:

$$E_{pq}^{t} = -f_{pq}^{t} * (\ln f_{pq}^{t}) \tag{3}$$

And the two-dimension entropy model of entire image t is:

$$E^{t} = \sum_{p=0}^{L-1} \sum_{q=0}^{L-1} E_{pq}^{t} \tag{4}$$

The two-dimension entropy model has two advantages. One advantage is that it can well depict the information contained in image, namely the complexity of image information, such as four images in Figure 1.

(a) (b) (c) (d)

Figure 1: One Advantage of Two-dimension Entropy Model: Image with More Disordered Content have Bigger Two-dimension Entropy Value. From Left to Right, 3.3188, 5.5439, 7.3011 and 8.8127 are the Corresponding Two-dimension Entropy Value. This Character Fits well our Intuitive Understanding.

Another advantage is that it emphasizes both the accumulated character of gray information distribution and corresponding averaging gray information in neighboring region of an image.

From the view of transition of shots, the visual content transition between shots in figure 1 can be seen as the cut transition. Besides cut transition, there exists gradual transition between shots: fade-in / fade-out and dissolve transition. Figure 2 (a) and (b) shows one example of fade-in / fade-out transition respectively.

(a): fade-in / fade-out transition

(b): dissolve transition

Figure 2: (a) A Compact Description of Fade-out and Fade-in Transition. (b) A Compact Description of Dissolve Transition.

In all, from what we have talked about, it is possible that using our proposed two-dimension entropy model can well complete the problem of shot segmentation and further cluster visually similar shot into a scene.

3. PROCEDURE OF SEGMENTATION

Let us recall what has been assumed in introduction that visual content often change from last past shot to the next one according to grammar of film language. That is, when visual contents between adjacent shots change, the clutter degree of shots is often different. Hence, the problem of segmentation of shot boundaries can change into another problem. Namely, looking for an appropriate measure to depict this clutter degree of image information and use this measure to separate temporal shots. Based on what has been discussed in section 2.2 in detail, we can see that two-dimension entropy model can well present clutter degree of image information and can be hence used to settle the problem of detecting shot boundary. A brief diagram of our proposed scheme is shown in Figure 3. Each component will be detailed one-by-one.

3.1 Detecting Shot Boundaries

Selecting location of shot break is the first step to be done for the task of the partition of temporal video shots. This process consists of following two steps. One is to detect the rough location of the shot transition by analyzing the property of frames within the sampling space. The other is to achieve the precise location of shot break by finding the frame with

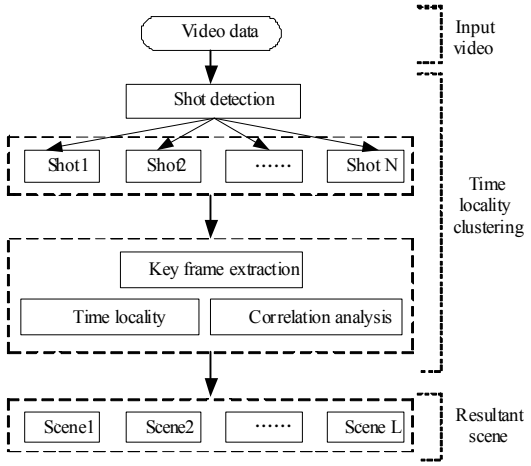


Figure 3: A Brief Diagram of Our Proposed Scheme.

minimum value of pixel dissimilarity within neighborhoods of the approximate location of a transition boundary.

Detecting Rough Boundary Point. During the course of the shot transition, the perceptually visual content between consecutive shots often changes. This change in visual content, here, can be well depicted using pixel dissimilarity based on our proposed two-dimensional histogram model described in the section 2. Let E_{ij}^{t} mean the two-dimensional entropy of the two-dimensional channel (i, j) in the t^{th} frame. Then the pixel dissimilarity between the t^{th} frame and its successor frame $t+1^{th}$ can be presented using the following function:

$$PD(t) = \sum_{p=0}^{L-1} \sum_{q=0}^{L-1} \min\left(E_{pq}^{t}, E_{pq}^{t+1}\right) \tag{5}$$

The candidate rough point of shot boundary can be declared if the following two criterion inequalities both satisfy condition, where T_{change} is a well defined threshold describing the important difference of two-dimensional statistic information between consecutive frames, and s is the sampling ratio:

$$\begin{cases} PD(t) - PD(t\text{-}1) < -T_{change} \\ PD(t+1) - PD(t) > T_{change} \end{cases} \tag{6}$$

Figure 4 show two examples of pixel dissimilarity of cut transition as shown in figure 1 and fade-out / fade in transition as shown in figure 2 (a) within the sampling space respectively. In our experiment, the sampling rate, T_{change} is set to be five.

Achieving Precise Transition Boundary. Once the rough position of shot transition is chosen, the precise candidate transition position can then be obtained by finding the frame whose pixel dissimilarity value is the minimum within the neighborhoods of the rough position in the original sequence space.

Suppose C is one detected approximate location of shot transition boundary. Then frame F can be considered as the candidate exact position of a transition boundary:

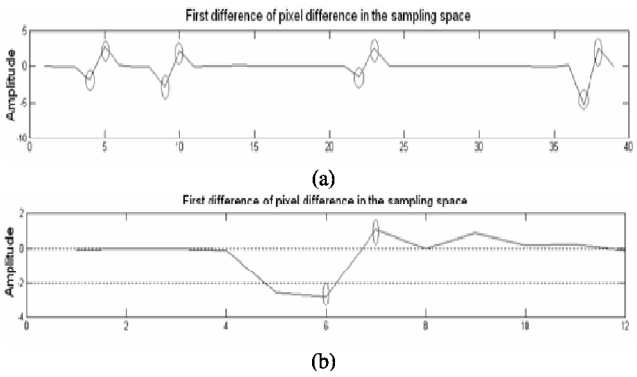


Figure 4: Show Examples of the First Difference of Pixel Dissimilarity within the Sampling Space of: (a) Cut Transition as shown in Figure 1; (b) Fade-out / Fade in Transition as shown in Figure 2 (a). The Red Circle Means the Local Minimum and Green Circle Means the Local Maximum of the First Difference of Pixel Dissimilarity within the Sampling Space.

$$F = \arg\min_{F} \left\{ PD(C * T_{length} - T_{length}), \ldots, PD(C * T_{length}), .., PD(C * T_{length} + T_{length}) \right\} \quad (7)$$

Figure 5 show the candidates precise transition location of cut transition as shown in figure 1 and fade-out / fade-in transition as shown in figure 2 (a).

Advantage of the Coarse-to-Fine Approach. It is necessary, here, to explain the reason why we use a coarse-to-fine approach rather than one direct pass to detect the candidate points of shot transition.

On the one hand, compared with the pixel dissimilarity of cut transition in the original space as shown in Figure 5 (a), it is easy to see that the direct approach of finding the local minimum to determinate the candidates of shot transition can result in many false positives for the gradual transition as shown in Figure 5 (b).

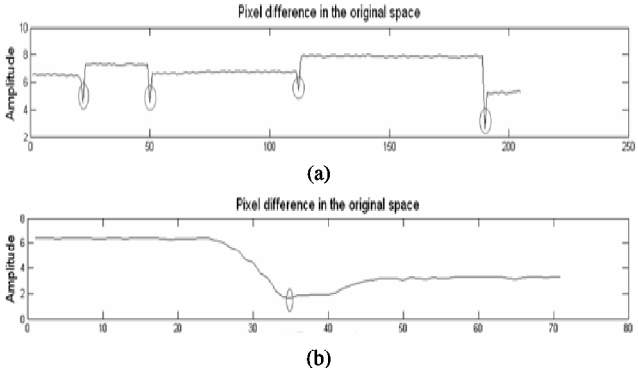


Figure 5: Show Examples of the Candidates Precise Transition Location Indicated by Red Circle of: (a) Cut Transition as shown in Figure 1; (b) Fade-out / Fade in Transition as Shown in Figure 2 (a).

On the other hand, from the comparative panes between Figure 6 (b) and Figure 6 (c), it is also clear to show that the advantage of using the difference of pixel dissimilarity in the sampling space rather than in the original video sequence to identify the boundary location of the type of gradual transition. That is, under the condition of using the difference of pixel dissimilarity between adjacent frames in the sampling space as the metric to detect the rough position of shot break, a delicate shot transition, such as the example of a video sequence shown in Figure 2 (a) may be missed. In addition, such disposal can efficiently avoid reducing a number of false alarms and the corresponding total processing time can be save largely, which can be justified from latter detection process.

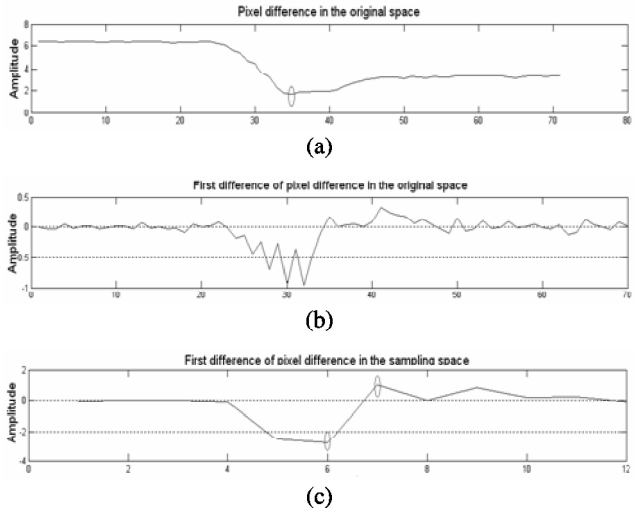


Figure 6: (a) Shows the Pixel Dissimilarity in the Original Video Space, (b) Shows the First Difference of the Pixel Dissimilarity in the Original Video Sequence, and (c) Shows the First Difference of the Pixel Dissimilarity in the Video Sampling Space.

Figure 7 shows some consecutive example shots detected using our algorithm described before in detail for the Hollywood cartoon movie "Ice Age II". This approach can determine the number of key frames according to the visual activity in corresponding shot, which will be represented in the following sub-section.

Figure 7: Some Consecutive Example Shots Detected using Our Algorithm for the Testing Hollywood Cartoon Movie "Ice Age II". Each Shot is Represented by One of the Key Frames.

3.2 Selection of Key Frame

In general, each shot can be depicted through a single key-frame, namely either the first frame, last frame or the middle frame. However, for some shots in feature film, the duration

time many be longer and camera motion may be faster. Hence, for such shots, the number of key-frames for each shot should coincide with the activity content. For example, for a shot on talk show, a single key-frame can just represent the typical meaning. While for a shot within high activity content, it needs a set of key-frames to sufficiently express the visual content. A large amount of techniques on key-frame selecting have been reported in [8]-[19]. In this paper, a two-pass algorithm is used to complete the task of the selection of key-frames. Given shot S, let b denote the beginning video frame, m denote the amount of frames and n denote the total of key-frames, the process of key-frames choosing can then be described as following several steps.

I: The first frame, b is chosen as the first key-frame K_l^s into the key-frame set K^s, n is correspondingly set to be 1. That is, $K^s = \{K_l^s\} = \{f_b\}$.

II: Starting from the first video frame f_i within given shot S, every key-frame within the key-frames set K^s is used to compute the value of histogram intersection between them. If all the values computed are greater than a pre-fixed threshold T_{change} the same threshold in the segmentation on video shots, then the current video frame f_i is absorbed into the key-frame set K^s as a new key-frame, and the total of key-frames, n is consequently added by 1. This step can be summarized as follow:

$$If \quad PD(f_i, K_n^s) > T_{change}, \quad \forall K_n^s \in K^s$$

$$Then \quad K^s = K^s \cup f_i, \quad n = n+1.$$

Figure 8 shows some examples of key frames detected in the testing Hollywood cartoon movie "Ice Age II". Specifically, key frames are selected from the example of detected shot in the figure 7. The resulting representative frames shows that the number of key frames is proportion to the shot visual activity in the corresponding shot.

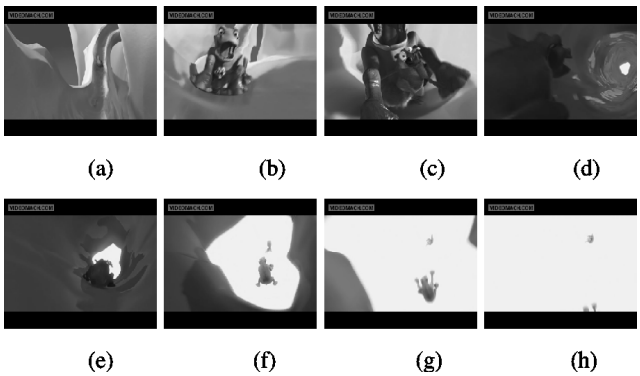


Figure 8: Key-frames on the Corresponding Example Shot in Figure 7 Shows that the Number of Key Frames is Proportion to the Shot Visual Activity in the Corresponding Shot. Image (a) is the one and only key frame selected from shot a in figure 7 (a), image (b) and (c) are the key frames to represent the visual content for shot as shown in figure 7 (b), the key frames of shot shown in figure 7 (c) consist of last five images in figure 8.

3.3 Similarity of Video Shots

According to encyclopedia definition, a scene can be understood as part of a story, which is a high-level temporal segment. A scene can be characterized either by the continuity in the visual contents of shots in which the setting is fixed, or by the continuity of the ongoing actions in the same locale. Namely, a scene is the set of shots with visually similar content. In addition, the duration of time of a scene complies with certain temporal constraint.

Similarity between two video shots should reflect the correlation between visual content elements, such as locales, persons and events, and so on. To match information between shots based on a similar way human deal with the problem of matching, it is more suitable to measure similarity between shots in terms of the intersection of the flow of imagery tracks. Such a similarity measure helps to group visually similar shots into a scene in a reasonable way. Otherwise, an unsuitable form of similarity measure will lead to cluster shots formerly belonging to the same scene into several different scenes.

A variety of criteria has been proposed to measure the dissimilarity between shots based on the respective key-frames only. Yeung *et al.* in [14] used information of color and luminance to measure the dissimilarity indices of video shots. Rasheed *et al.* in [21] used the Backward Shot Coherence (BSC) method to measure the similarity for a given shot with respect to the previous shots. Niblack *et al.* in [22] proposed the method of determination of quadratic distance between color histogram to measure similarity between images. Strickter *et al.* in [23] presented a similarity measure by computing the distance based on low-order moments of histogram to cluster similar shots into scene. Zhai *et al.* in [24] computed the similarity in terms of the Bhattacharya distance. In this paper, we propose a suitable inter-shot similarity measure based on the reciprocal of the Bhattacharya distance to measure the similarity between shots. The definition of similarity between two video shots consists of two steps described as follows.

I: Given two shots $SB_{1B} = KP^{s1\,P} = \{KB_{j1PB}^{S1P}\}\ B_{j1=1PB}^{n1P}$ and $SB_{2B} = KP^{s2\,P} = \{KB_{j2PB}^{S2P}\} B_{j2=1PB}^{n2P}$, SB_{1B} and $SB_{2\,B}$ are similar if a pair of key-frame KB_{jPB}^{S1P}, KB_{jPB}^{S2P} is similar, where nB_{1B} and nB_{2B} are total number of key-frames in corresponding shot.

II: The similarity between two video shots is here measured based on the reciprocal of the Bhattacharya distance, where jB_{1B} and $jB_{2\,B}$ are the key frame from different shot SB_{1B} and $SB_{2\,B}$ respectively:

$$RBD(j_1, j_2) = -1 \Big/ \left(\ln \left[\sum_{p=0}^{L-1} \sum_{q=0}^{L-1} \sqrt{E_{pq}^{j_1} * E_{pq}^{j_2}} \right] \right) \tag{8}$$

The formulation of similarity between shot SB_{1B} and SB_{2B} is:

$$\begin{aligned} ShotSim(S_1, S_2) &= \max\left(RBD\left(K^{S_1}, K^{S_2} \right) \right) \\ &= \max\left(RBD\left(\left\{ K_{j_1}^{s_1} \right\}_{j_1=1}^{n_1}, \left\{ K_{j_2}^{s_2} \right\}_{j_2=1}^{n_2} \right) \right) \end{aligned} \tag{9}$$

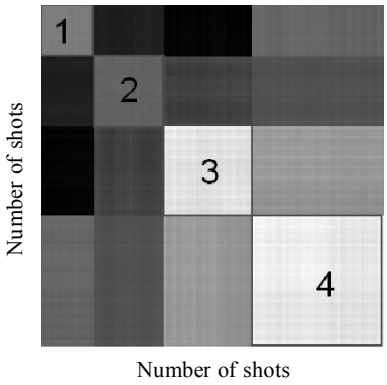


Figure 9: A Visual Shot Similarity Map of the First Four Pairs of Shots for "Ice Age II". Shots from the Same Scene form a Bright Block along the Diagonal. Brighter Block Represents Higher Similarity. The Ground Truth Scene Boundaries are Indicated by Rectangle Solid Lines.

Figure 9 shows the visual similarity map of the first four pairs of shots of one testing Hollywood cartoon movie "Ice Age II". The similarity between shots is depicted by pixel intensities such that two interesting facts can be easily found. On the one hand, the brighter block means higher similarity. That is, shots from the same scene form a bright block along the diagonal. On the other hand, there is an obvious border between different blocks indicated by rectangle solid lines in corresponding scenic region. Shots shown in figure 7 are chosen from the same scene "Looking for the acorn"; hence they fall into the same bright block annotated with number 1.

3.4 Analysis of Time Constraint

For the purpose of segmentation of video scene boundary in a semantic way, it is necessary to introduce a temporal locality constraint. The reason for using the technique of time constraint during the process of grouping visually similar shots into the same scene is to keep from under-segmenting. Specifically, if temporal interval between two shots exceeds certain constraint, then although their visual content is similar, they can not be grouped into the same scene yet. In fact, the similarity between them is only in visual aspect, the factual content in terms of human understanding is sufficiently different. Or the two shots are recorded at different scenes.

Figure 10 shows some example shots extracted from different scenes of the movie "Ice Age II" to explain such two case of visual similarity. The visual concept of top three corresponding shots is similar, however their factual content are distinct from each other. As a matter of fact, they are selected from the annotated scene "talking about ice dissolve", "fighting with big fish", "recalling her past" respectively. While for the bottom two shots, although they are recorded in the similar location, the corresponding episode takes place in different scenes. The former is in the annotated scene "talking about ice dissolve", while the latter is from the annotated scene "fighting with flood".

Figure 10: Some Example Shots Represented by One Representative Frame are Extracted from Different Scenes of the Hollywood Cartoon Movie "Ice Age II" to Illustrate the Importance of Time Constraint. For the Top Three Shots, Although They have Similar Visual Content, their Factual Content is Distinct. While for the Bottom two Shots, Although they are Recorded in the Similar Location, the Corresponding Episode Takes Place in Different Scenes.

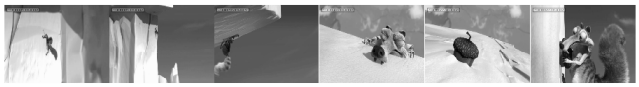
For the technique of temporal constraint clustering, the parameter T_{window} is introduced to denote the time-window length. Here, time-window length T_{window} shows the maximum of total number of shots a scene can contain. It means that only if shots falling within this time-window length and satisfying certain criterion of clustering discussed in section 3.5, they can be grouped into the same scene.

The value of time-window length T_{window} should be selected to reflect reasonable time duration of a scene and segment video stream file into approximate actual scenes from the perspective of human perceptivity. Accordingly, the choice of time-window length T_{window} greatly affects the finally resulting scene boundaries. On the one hand, if this value is chosen to be very large, it will then cover more than one scene. It means that there may be more number of shots compared with within a potential scene and subsequently result in under-segmentation. On the other hand, if this value is too small, the number of shots will be less than that within a potential scene and cause over-segmentation. Let us take the testing movie "Ice Age II" to explain the important of the choice of the value of the time-window length.

Figure 11 shows some detected examples shots which is depicted by one of the key frames in respective shot. The top shots are selected from the scene "telling a story" and the middle shots are extracted from the scene "talking about doomsday". These two scenes are both interacting events

(a): Typical shots from the scene 'tell a story'.

(b): Typical shots from the scene 'talk about doomsday'.



(c): Typical shots from the scene 'Looking for the acorn'.

Figure 11: Illustrate the Importance of the Choice of the Value of Time Constraint

where shots interact between Dieco and animals. The shot of Diego is shown in both two scenes such that the time-window length with very big value will lead to under-segmentation. For the bottom shots extracted from the scene "Looking for the acorn", which describe a serial event without any interaction shot. In such case, if the value of the time-window length is selected too small, then the resulting segmentation of scenes will be of over-segmentation.

In our experiment, time-window length T_{window} is set to be thirty. The reason why the value of time-window length T_{window} is set to be thirty is discussed in section 4 in detail.

3.5 Clustering of Video Shots

In this section, in view of the clustering from the semantic level, shots with visually similar content should be clustered into the same scene based on the similarity measure and temporal constraint. Meanwhile, shots from different scenes should have sufficiently different visual content and accordingly should not be grouped into the same cluster by mistakenly. The process of grouping consists of following several steps.

I: The similarity between two key frames j_1 and j_2 from different shots S_1 and S_2 is measured by the reciprocal of the Bhattacharya distance:

$$RBD(j_1, j_2) = -1 \Big/ \left(\ln \left[\sum_{p=0}^{L-1} \sum_{q=0}^{L-1} \sqrt{E_{pq}^{j_1} * E_{pq}^{j_2}} \right] \right) \quad (10)$$

II: The measurement of the dissimilarity between shot S_1 with n_1 key frames and S_2 with n_2 key frames is the maximum across all pairs of frames in each shot:

$$ShotSim(S_1, S_2) = \max \left(RBD \left(\left\{ K_{j_1}^{s_1} \right\}_{j_1=1}^{n_1}, \left\{ K_{j_2}^{s_2} \right\}_{j_2=1}^{n_2} \right) \right) \quad (11)$$

III: Within one time-window T_{window}, we compute the shot similarity between all pairs of shots S_i and S_j:

$$SceneSim(T_{window}) = ShotSim(S_i, S_j),\ i, j \in (1, T_{window}) \quad (12)$$

IV: For the current shot *Sc* within T_{window}, finding all the subsequent shots sharing similar visual content and picking out furthermost two shots S_{f2} and S_{f1} for subsequent processing:

$$S_f = ShotSim(S_c, S_i) > T_{scene}, i \in (c, T_{window}) \quad (13)$$

where T_{scene} is the fixed-value to group similar shots into the same scene. Furthermore, S_{f1} is chosen as the current shot *Sc* for subsequent processing.

V: For shot *Sc* between S_{f2} and S_{f1}, finding whether there exist similar shots between S_{f1} and T_{window}, and picking out furthermost two shots S_{b2} and S_{b1}:

$$S_b = ShotSim(S_c, S_i) > T_{scene}, c \in (f_2, f_1), i \in (f_1, T_{window}) \quad (14)$$

Furthermore, S_{b1} is also chosen as the current shot *Sc* for subsequent processing.

VI: Repeat step 4 or / and 5 until the end of T_{window}. If the iterative process stops at step 4, then shot S_{f1} is the boundary of current scene and S_{f1+1} is the beginning shot for next scene; if the iterative process stops at step 5, then shot S_{b1} is the boundary of current scene and S_{b1+1} is the beginning shot for the next scene.

Figure 12 shows the process of one scene boundary detection for 'Ice Age II' within one time-window. For shot S_{f3}, S_{f2} and S_{f1} are the similar shots during the forward searching for S_{c1}. For shot S_{c2} between S_{f2} and S_{f1}, S_{b1} is the similar shot. There are no similar shots to shot S_{b1}; hence the current scene S_{b1} is composed of shots from S_{c1} to S_{b1}.

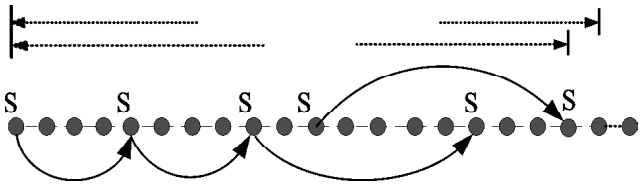


Figure 12: Example of the Process of the Determination of Scene Boundary using Special Correlation between Shots under the Condition of Temporal Constraint

4. EXPERIMENTAL RESULTS

This section describes experimental results of the analyses applied to three test feature films and one TV interview. The results of the segmentation of original video sequences into semantic scene units are based on the temporal locality constraint and shot coherence comparison discussed detailed in section 3.4 and 3.5. All the experiments are done on the Intel Pentium 3.0 GHz machine running Windows. Performance measures and matching rule of scenes are presented in section 4.1. Section 4.2 shows the characteristics of test feature films. Two important parameters, T_{window} and T_{scene} used in the procedure of grouping similar shots into the same scene, are discussed in section 4.3. The best two values are experimentally determined. This is done by fixing one value of parameter and varying the other parameter value on feature film Ice Age II.

4.1 Performance Measures and Matching Rule

The system performance of our proposed method is measured by following two metrics: *precision* and *recall*:

$$\text{Precision} = \frac{N_c}{N_d}, \quad \text{Recall} = \frac{N_c}{N_g} \quad (15)$$

where N_c is the total number of correctly detected shots, N_d is the total number of detected shots, and N_g is total number of ground truth shots. From the perspective of system performance, high precision is pleasing for it shows correct scene boundaries are furthest detected. Similarity, high recall is desirable based on the following fact. High recall describes most scene boundaries can be uncovered by human knowledge.

For a given feature film, suppose N_g is the total number of ground truth scene boundaries and $Sg = \{Sg_{1b}, Sg_{1e}, \ldots Sg_{Ngb}, Sg_{Nge}\}$ be the number of beginning shot and ending

shot in corresponding scene. Correspondingly, let N_d be the total number of scene boundaries detected by our proposed shot clustering method and $Sd = \{Sd_{1b}, Sd_{1e}, \ldots, Sd_{Ndb}, Sd_{Nde}\}$ denote the number of beginning shot and ending shot in corresponding scene. In theory, if one pair of shot boundaries Sd_{ib}, Sd_{ie} in testing set has same beginning shot and ending shot Sg_{jb}, Sg_{je} in ground true set, Sd_i is then declared as the correct scene boundary. However, taking into account of editing effect, another alternative rule is used in the procedure on practical matching. That is, if more than 80% shots in testing scene Sd_i and ground truth scene Sg_j overlap each other, Sd_i and Sg_j can then be said to match.

4.2 Characteristics of Test Feature Films

In our experiment, three entire MPEG-1 feature films, namely Ice Age II (I. A. II), Mission Impossible III (M. I. III) and X Man III (X M. III) and one TV interview (TV Inter.). Frame rate of each video stream file is 25 frames/s and the corresponding spatial resolution is 320□–240 pixels. Table 1 tabulates the detail characteristics of the testing videos. In addition, the ground truth scenes in this experiment are achieved by manually segmented from original video files.

Table 1
Detail Characteristics of the Three Test Videos

Name	*TV Inter.*	*I. A. II*	*M. I. III*	*X M. III*
Time	00:39:23	1:20:02	1:02:09	1:44:03
# Frames	59097	121557	150808	156084
# Shots	422	1240	1809	1219
# Scenes	15	35	45	25

4.3 Variation of Important Clustering Parameters

There are three important parameters in our experiment, namely parameter about video shots detection T_{change}, parameter on time-window length T_{window}, and parameter of video shots clustering T_{scene}. The process of choosing best values of three important parameters introduced above is described as below.

The parameter T_{change} are achieved from multi-time experimental results and chosen as 0.1 respectively, resulting in good performance on video shots detection and corresponding key-frames selection. As for parameters T_{window} and T_{scene}, the procedure of choosing best values is experimentally determined. The best parameters values chosen can offer good trade-off between precision and recall. Since there is no simple theoretically correct way to choose the value on time-window length and video shots clustering, several cases of different values are tried.

Table 2 shows the result of testing video 'Ice Age II' under the condition of T_{scene} with fixed value 0.35 and T_{window} with various values within the range from 20 to 60.

Table 2
Detection Results of Testing Video 'Ice Age II' under the Condition of Different T_{window} Values and Fixed T_{scene} value-0.35

Measure	*Time-window length value (T_{window})*				
	20	*30*	*40*	*50*	*60*
# Detected scenes	70	45	40	37	35
# Matched	22	24	25	31	27
# Missed	13	11	10	4	8
Precision	0.31	0.53	0.63	0.84	0.77
Recall	0.63	0.69	0.71	0.89	0.77

Table 3 gives the results testing video 'Ice Age II' using various values on video shots clustering from 0.25 to 0.45 and a fixed value on time-window length 50.

Table 3
Detection Results of Testing Video 'Ice Age II' under the Condition of Various TBscene Values and Fixed TBwindow Value-50

	Video shots clustering value (TB_{sceneB})				
	0.25	*0.30*	*0.35*	*0.40*	*0.45*
# Detected scenes	48	43	37	35	32
# Matched	25	26	31	25	24
# Missed	7	6	4	7	9
Precision	0.52	0.60	0.84	0.71	0.75
Recall	0.71	0.74	0.89	0.71	0.69

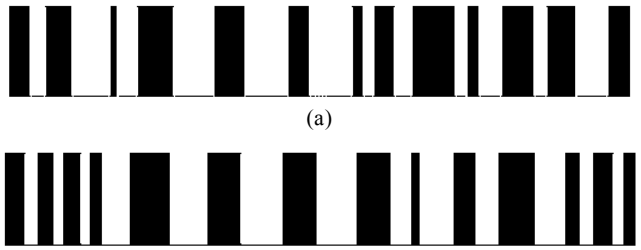
From above detail discussion, it can be seen that the best performance for the task of the segmentation on scene boundaries is achieved when the value on time-window length T_{window} and video shots clustering T_{scene} are 50 and 0.35 respectively. Thus, (T_{window}, T_{scene}) of (50, 0.35) is selected as the best parameter value for scene boundary detection.

Table 4 gives the final detection results with the best parameter value 50 time-window length and 0.35 video shots clustering. From this table, we can see that the overall precision and recall accuracy are 79% and 86.1%, which fits evaluation of system performance and good balance between precision and recall.

Table 4
Detail Detection Results for Scene Boundaries Segmentation with the Value Time-window Length of 50 and the Value Video Shots Clustering of 0.35, where # D Scenes Stands for the Total Number of Detected Scenes

Measures	*TV Inter.*	*I. A. II*	*M. I. III*	*X M. III*
# Scenes	15	35	45	25
# D Scenes	16	37	52	27
# Matched	13	31	40	20
Precision	0.813	0.838	0.769	0.741
Recall	0.867	0.886	0.889	0.800

Figure 13 shows the detail scene detection results of the movie 'X Man II'. The upper row indicates the ground truth scenes represented by alternating black / white stripes, and the bottom row is the detected results using the proposed scheme.



(a)

(b)

Figure 13: Shows the Ground Truth Scenes and Detected Scenes of the Movie 'X Man III', Where Consecutive Scenes are Represented by Alterative Black / White Stripes. The Bottom Row Shows the Scenes based on the Consistent Observation Result of Ten Human Observes, and Detected Scenes using the Proposed Scheme are shown in the Upper Row.

We compare the system performance of scene segmentation between the results generated by the proposed algorithm and the Rui method [25], which is shown in Table 5. From the table5, it can be observed that the average precision and recall of our approach are and respectively, while the average precision and recall of [25] are and respectively.

Table 5
Comparison with Content-of-Table [25]

Name	*TV Inter.*	*I. A. II*	*M. I. III*	*X M. III*
Proposed method				
Precision	0.813	0.838	0.769	0.741
Recall	0.867	0.886	0.889	0.800
TOF				
Precision	0.728	0.742	0.712	0.676
Recall	0.806	0.645	0.763	0.638

5. DISCUSSION AND CONCLUSION

In this paper, a new method is presented to complete the temporal scene segmentation of video track. The detected scenes can closely approximate factual video episodes. Our segmentation is based on the investigation of the visual information between adjacent scenes, as well as on the assumption that the visual content within a scene is similar. Consequently, the two-dimension entropy model is used to depict the representative character for different scenes. To depict the content of a shot more truly, appropriate amounts of key-frames are select based on the activity content within relevant shot. Using temporal constrained clustering and forward shot coherence comparison analysis can efficiently group shots with similar visual content into the same scene and avoid occurring the phenomenon of under-segmentation. Based on experimental results, a following phenomenon can be easily discovered. Although the three test feature films belong to quite different categories, the final detected results do not differ much, which shows our propose system fits for different types of feature films. Namely, with a common set of parameters, we can achieve satisfactory and meaning scenes, presenting distinct events and locales from different categories of video programs. From table 4, we can see high accuracy measures can be obtained. In addition, the final resulting scenes can be used to solve the movie retrieval task by developing an efficient blue print on event-driven video retrieval.

REFERENCES

[1] V. Kobla, D. DeMenthon, and D. Doermann, "Special Effect Edit Detection using Video Trails: A Comparison with Existing Techniques", in Proc. on Conf. Storage Retrieval Image Video Databases, 1999, 302–313.

[2] N. Dimitrova, H. J. Zhang, and *et al.*, "Applications of Video Content Analysis and Retrieval", *IEEE Trans*. Multimedia, **9**(3), 2002, 42–55.

[3] A. Hanjalic, "Shot-boundary Detection: Unraveled and Resolved?", *IEEE Trans. Circuits System and Video Technology*, **12**(2), 2002, 90–105.

[4] Z. Cernekova, C. Kotropoulos, and I. Pitas, "Video Shot Segmentation using Singular Value Decomposition", in *Proc. on IEEE Int. Conf. Multimedia and Expo*, 2003, 301–302.

[5] M. Albanese, A. Chianese, and *et al.*, "A Formal Model for Video Shot Segmentation and its Application via animate Vision", *Trans. Multimedia Tools and Application*, **24**(3), 2004, 253–272.

[6] G. Boccignone, A. Chianese, and *et al.*, "Foveated Shot Detection for Video Segmentation", *IEEE Trans. Circuits System and Video Technology*, **15**(3), 2005, 365–377.

[7] J. Yuan, J. Li, F. Lin, and B. Zhang, "A Unified Shot Boundary Detection Framework Based on Graph Partition Model", in Proc. on ACM Multimedia, 2005, 539–542.

[8] Andreas Girgensohn, and John Boreczky, "Time-Constrained Key frame Selection Technique", *Trans. Multimedia Tools and Applications*, **11**(3), 2000, 347-358.

[9] W. Ren, and S. Singh, "TA Novel Approach to Key-frame Detection in Video",T in Proc. on Visualization, Imaging, and Image Processing, 2003.

[10] Satoshi Hasebe, Makoto Nagumo, and *et al.*, "Video Key Frame Selection by Clustering Wavelet Coefficients", in Proc. on European Signal Processing Conference, 2004, 2303-2306.

[11] Hideyuki Togawa, and Masahiro Okuda, "Position-Based Key Frame Selection for Human Motion Animation", in Proc. on Parallel and Distributed Systems, 2005.

[12] Thorsten Thormaehlen, Hellward Broszio, and Axel Weissenfeld, "TKey Frame Selection for Camera Motion and Structure Estimation from Multiple Views", in Proc. on Europe Conference Computer Vision, 2004, 523–535.

[13] Z. Xiong, X. Zhou, and *et al.*, "Semantic Retrieval in Video", *IEEE Trans. Signal Processing*, **23**(2), 2006, 18-27.

[14] Minerva Yeung, Boon Lock Yeo, and Bede Liu, "Segmentation of Video by Clustering and Graph Analysis", in Proc. on Computer Vision and Image Understanding, 1998.

[15] John S. Boreczky, and Lynn D. Wilcox, "A Hidden Markov Model Framework for Video Segmentation using Audio and Image Features", in Proc. Int. Conf. Acoustics, Speech, and Signal Processing, 1998.

[16] Hari Sundaram, and Shih-Fu Chang, "Video Scene Segmentation Using Video and Audio Features", in Proc. on IEEE International Conference on Multimedia and Expo, 2000, 1145-1148.

[17] Zeeshan Rasheed, and Mubarak Shah, "A Graph Theoretic Approach for Scene Detection in Produced Videos", in Proc. on ACM Workshop Multimedia Information Retrieval, 2003.

[18] Wallapak Tavanapong, and Junyu Zhou, "Shot Clustering Technique for Story Browsing", *IEEE Trans. Multimedia*, **6**(4), 2004, 517-526.

[19] Hao Jiang, Tong Lin, and Hongjiang Zhang, "Video Segmentation with the Support of Audio Segmentation and Classification", in Proc. on IEEE Int. Conf. Multimedia and Expo, 2000.

[20] Daniel Arijon, Grammar of Film Language, Hasting House, NY, 1976.

[21] Zeeshan Rasheed, Mubarak Shah, "A Graph Theoretic Approach for Scene Detection in Produced Videos", in Proc. on Computer Vision and Image Understanding, 2003.

[22] W. Niblack, R. Barber, and *et al*, "The QBIC project: Querying images by content using color, texture and shape", in Proc .on Storage and Retrieval for Image and Video Databases, 1993, 13–25.

[23] M. Strickter and M. Orengo, "Similarity of Color Images", in Proc. on Storage and Retrieval for Image and Video Databases, 1995, 381–392.

[24] Yun Zhai, and Mubarak Shah, "A General Framework Temporal Video Scene Segmentation", in Proc. on IEEE International Conference on Computer Vision, 2005.

[25] Y. Rui, T. S. Huang, and S. Mehrotra, "Constructing table-of-content for videos", ACM Multimedia Systems Journal, Special Issue on Multimedia Systems on Video Libraries, **7**(5), 1999, 359-368.