Temporal Multi-Resolution Framework for Shot Boundary Detection and Keyframe Extraction

Chandrashekhara A, HuaMin Feng and Tat-Seng Chua
School of Computing, National University of Singapore
Email: {chuats,fenghm,chandra}@comp.nus.edu.sg

Abstract

Video shot boundary detection and keyframe extraction is an important step in many video-processing applications. We observe that video shot boundary is a multi-resolution edge phenomenon in the feature space. In this experiment, we expanded our previous temporal multi-resolution analysis (TMRA) work by introducing the new feature vector based on motion, incorporating functions to detect flash and camera/object motion, and selecting automatic thresholds for noise elimination based on the type of video. The framework is used to extract meaningful keyframes. Experiments show that our new system can detect and characterize both the abrupt (CUT) and gradual (GT) transitions effectively. It has good accuracy for both the detection of transitions as well as their boundaries.

1. Introduction

Due to the presence of many types of transitions and the wide varying lengths of GTs, the task of detecting the type and location of transitions in video is a complex task. In fact, the transition of video is not a single resolution phenomenon. For example, although longer GT can't be observed at a high temporal resolution, it is apparent at a low temporal resolution of the same video stream. Thus the detection of transitions of video shots is a temporal multi-resolution problem. Information across resolutions can also be used to detect as well as locate both the CUT and GT transition points. Since wavelet is well known for its ability to model sharp discontinuities and to process signals according to scales [1], we employ Canny-like B-Spline wavelets in this multi-resolution analysis. This work provides: a) an unified approach for CUT and GT detection; b) accurate location of gradual transition boundary; c) adaptive threshold value selection based on video variance within a sliding window; d) flash elimination by characterizing the phenomenon in multi resolution; e) motion elimination by computing quadratic similarity measures within the transition and its neighborhood; and f) keyframe extraction.

2. Basic Theory

2.1 Video Representation

We model the video according to the content of the video frames in the stream. The feature for representing the content of video frames could be of any type: color, shape, texture or motion. Thus video is modeled in a N-dimensional feature space of : (a) gray-level representation, (b) RGB value, and (c) Optic flow/motion vector value. The dimension of the space depends on the dimensionality of the chosen features. Since the color-histogram representation has been found to be useful for the video segmentation problem, we use the N-color histogram for each frame of video. Our experiment shows that the local histogram-based method has difficulties in improving both the recall and precision of the shot boundary detection at the same time. In addition it has difficulty locating precise boundary of GT due to the flash and camera/object motions. To overcome this problem we constructed a motion-based feature using the motion-vectors of MPEG compressed stream. Besides the use of these features, we also use derivatives to detect the transitions. The maximas of the first order derivative or zero crossing in the second order derivative will correspond to transition points. In this paper, the first order derivative was taken for easier implementation.

By empirically observing GTs that exists in most video streams, we find that different types of GTs exist like fade in/fade out, dissolve, wipe, morph etc. Moreover, the length of the transition may vary greatly too. Different shot transitions have different characteristics, so it is hard to use just one single feature and single algorithm to capture the characteristics of all kinds of shot transitions efficiently. Just as the assumptions most existing algorithm follows, one can clearly observe that the content between shots change much more than intra-shot change. However different types of shot transitions are observable at different scales in the feature space. Whatever the type or length of the transition, there will always be a change big enough that we can detect. The difference is

only the resolution of our observation. For CUT, we could see the change both in a detailed observation (between two successive frames), or a coarse observation (across several frames), while GT only shows the change in a coarse observation. So the transition must be defined with respect to different resolutions. By viewing the video at multiple resolutions, the CUT and those GTs could be unified. The only difference is that GTs means boundaries of signal in low resolutions while CUT means in all the resolutions. By making this fundamental observation that a video shot boundary is a multi-resolution phenomenon, we can characterize the transitions with the following features: the scale of the transition, the strength of the transition, and the singularity of the transition point. We have developed a unique multi-resolution analysis technique to detect and characterize both the CUT & GT shot boundaries.

2.2 Applying Wavelet

The multi-resolution phenomenon has been widely studied in other areas, and wavelets provide a good mathematical basis for such an analysis. In the analysis, we need to construct a scale space. The Gaussian scale-space approach is widely adopted as the Gaussian function is the unique kernel, which satisfies the causality property as guaranteed by the scaling theorem. Because the first order derivative of the Gaussian function could be a mother wavelet, one can easily show that the sharper variation points of the signal corresponds to the local maxima of the wavelet transform. Thus a maxima detection of the wavelet transform is equivalent to boundary detection. If the mother wavelets is the Canny wavelet, which is the first order derivative of the Gaussian , then

$$\psi^a(x) = \frac{d\theta}{dx} \tag{1}$$

and the dilation at scale is

$$\psi_s^a(x) = \frac{1}{s}\psi^a(\frac{x}{s}) \tag{2}$$

When choosing a dyadic scale sequence 2^j , we get:

$$\psi_{2^j}^a(x) = \frac{1}{2^j}\psi^a(\frac{x}{2^j}) \tag{3}$$

The wavelet transform here is defined as:

$$W_s^a f(x) = f \times \frac{1}{s}\varphi^a\left(\frac{x}{s}\right) = \frac{1}{s} f \times \frac{d}{d\frac{x}{s}}\theta\left(\frac{x}{s}\right) = s\frac{d}{dx}[f \times \theta_s(x)] \tag{4}$$

From the right side of equation (4), we can see that the resulting output is a smoothed signal generated by a Gaussian filter that calculates the first order derivatives. It could be shown here that this wavelet transform is equivalent to smoothening the signal by applying the different scale Gaussian filters and then calculate the first order derivatives. The detailed derivation of Equations (1-5) can be found in [3]. The local maximas of the resulting signal will indicate where the transitions happen, and the magnitude of the maximas will show the strength of the transitions. Tracing the maximas in different resolutions is equivalent to finding transition points in different resolutions. In many cases, the presence of noise may result in maximas too. We distinguish the real transitions from the noise by examining the cross-resolution information. A real transition will still be a maxima in all resolutions. However, the noise may be lost or eroded in a lower resolution of the smooth function [3].

3. Implementation

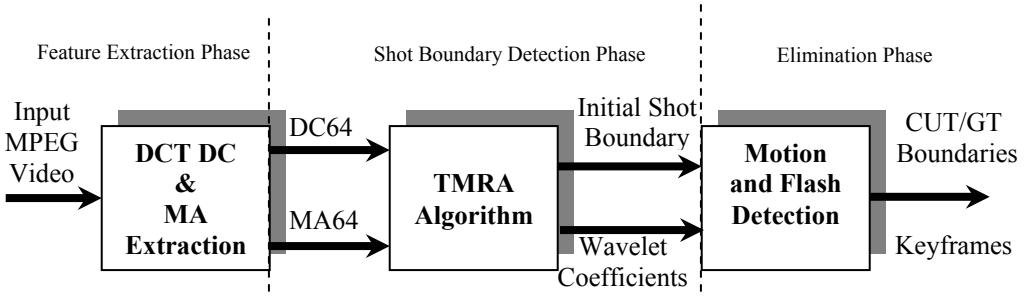

Figure 1. TMRA system for shot boundary detection

The TMRA system has 3 phases. In the feature extraction phase, feature vectors suitable for the TMRA are computed. In the shot boundary detection phase, transitions are characterized and the TMRA algorithm is applied to obtain the transition boundaries. This result will include many wrong transitions (insertions). In the elimination phase, motion analysis and flash detection are applied to remove insertions due to motion and abrupt flash noises. Figure 1 shows the system architecture for shot boundary detection. The following sections discuss in detail each of these three phases.

3.1 Feature extraction

We extract motion vector feature and the color histogram feature. The DC64 color histogram is computed by extracting the DCT DC value for each block in the frame. The value is quantized to 64 values. The MA64 direction histogram is computed using the motion vectors for each macro block and quantizing the angle values to 60 bins. Since the motion vectors tend to be sparse, a 3X3 median filtering is applied to the motion vectors. Also boundary blocks are not considered in the formation of the feature vectors, as they tend to be erroneous. The last 4 bins contain the macroblock type count of forward predicted, backward predicted, intra and skip macroblocks.

3.2 TMRA Algorithm

In this phase, TMRA algorithm is applied to determine the potential transition point and their type. It performs the wavelet transformation on the color and motion based feature vectors.

3.2.1 Locating potential transitions

The goal of video segmentation is not only to detect the occurrence of a transition, but also to locate the exact positions of the CUT/GT to segment the video. In a GT, both the start position and end position need to be detected. Low resolution of the wavelet coefficients helps to detect the occurrence; where as high resolution helps to characterize the start and end of the transition. In the higher resolution, the boundaries would show up as the maxima points. To identify this boundary we use both the DC64 and MA64 wavelet coefficients. For low resolution (Resolution 3) DC64 wavelet coefficients are used and for high resolution (Resolution 0) MA64 wavelet coefficients are used. The DC64 feature space fails to characterize the beginning and ending frames of the transitions accurately at the high resolution. This is due to the fact that the rate of change in DC64 feature space is not high and doesn't result in a distinguishable peak. Hence we designed a new feature space based on direction of motion along with counts representing static (skip) and intra (blocks having significant change) blocks. As a result of this we observe that even a gradual change resulted in an abrupt spike (peak) at the start and end of the transition. This is captured as the boundary of the transition. After we identify the local maxima points at some lower resolution (also called potential transitions), we use these local maxima points as the anchor points, and trace up to the higher resolution of the motion-based wavelet coefficients.

3.2.2 Adaptive thresholding

The problem of choosing appropriate threshold is a key issue in applying TMRA for shot boundary detection. Heuristically chosen global threshold is not suitable as the shot content changes from scene to scene. So adaptive thresholds are better than a simple global threshold. Here we use a sliding window method to calculate the thresholds. Our system has one weighting factor which can adaptively adjust based on the sliding window size and the standard deviation of dc64 feature of the neighborhood frames.. For different video clips, their standard deviations (see equation 5) are different. The standard deviation of home videos is larger than the general videos. The choice of sliding window size is also very important. In our system, we scan the whole wavelet coefficients of DC64 and choose the sliding window size as the sum of max interval of peak points and max interval of valley points (see equation 6). This removes most of the noise peak points due to brightness/contrast variations, blurring and small motions. The weighting factor can be used to adjust the threshold, which is chosen in the range of 0.8~1.5. In general, for home video, the weighting factor can be larger, in the range of 1.2~1.5, whereas for other video, it can be in the range of 0.8~1.1.

$$S = \sqrt{\frac{1}{N-2}\left(\sum_{i=1}^{N}(f_i - f_{i-1})^2 - \frac{\left(\sum_{i=1}^{N}|f_i - f_{i-1}|\right)^2}{N-1}\right)} \tag{5}$$

where N denotes the total frame number of the video sequence; f denotes the feature (DC64).

$$dis_p = \arg\max_{i \in N_p}\{D_i^{(p)}\}, dis_v = \arg\max_{i \in N_v}\{D_i^{(v)}\}$$
$$size = dis_p + dis_v \quad (6)$$

where Np denote the number of the peak points and Nv denote the number of valley points. $D_i^{(p)}$ is the interval of neighborhood peak points whereas $D_i^{(v)}$ is the interval of neighborhood valley points. dis_p and dis_v are maximum intervals of peak points and valley points, respectively.

3.3 Elimination and Keyframe Extraction

The last phase of the TMRA algorithm is the elimination of wrong transitions due to motion and abrupt flashes in the shot. Also representative keyframes are extracted for each shot. The following sections give a brief description on the method to characterize such activities in the multi resolution framework.

3.3.1 Flash detection

With our TMRA, flash is easily detected by testing the changes of frame's wavelet coefficients at all resolutions. For abrupt noise, the magnitude of coefficient value also decreases as the resolution decreases.

3.3.2 Camera/Object motion detection

With our ATMRA, we detect camera and object motion points. In principle, for the correct transitions (CUT/GT), the mean absolute differences (MAD) of DC64 and MA64 should be consistent. At CUT and GT points, the MADs are consistent across both the DC64 and MA64 feature space. If MAD changes are not consistent across DC64 and MA64 around the potential transition points, then it is likely to be a wrong transition caused by the camera/object motion.

Summarizing our motion detection method: first we compute three kinds of quadratic differences (distance between mean absolute difference of feature vector) for each potential transition we have found. They are represented as QMADbefore(similarity before the transition), QMADintra (similarity within the transition), QMADafter (similarity after the transition) as shown in Equations (7-9). Here $C_k^r = \frac{k!}{r!(k-r)!}$ is the normalizing factor. For each transition, we compute these three parameters for DC64 and MA64 feature, respectively.

$$QMAD_{before} = \left(\sum_{i=start-k}^{start-1} \sum_{j=i+1}^{start} | f_i - f_j | \right) / C_k^2 \quad (7)$$

$$QMAD_{inter} = \left(\sum_{i=start}^{end-1} \sum_{j=i+1}^{end} | f_i - f_j | \right) / C_{end-start+1}^2 \quad (8)$$

$$QMAD_{after} = \left(\sum_{i=end}^{end+k-1} \sum_{j=i+1}^{end+k} | f_i - f_j | \right) / C_k^2 \quad (9)$$

where: start and end represents the begin and end frame number of the potential transition. k represents the computing range ($2 \le k \le 9$). fi denotes the feature.

We remove those potential transitions that meet the following condition:

$$\left(\begin{array}{l} \left(QMAD_{inter}^{(dc)} \ge \left(QMAD_{before}^{(dc)} + QMAD_{after}^{(dc)}\right)/2\right) \wedge \\ \left(QMAD_{inter}^{(mv)} < \left(QMAD_{before}^{(mv)} + QMAD_{after}^{(mv)}\right)/2\right) \end{array} \right) \quad 10)$$

3.3.3 Keyframe extraction

Keyframes are very useful to summarize videos, and to provide access points into them. In this paper we also derive distinct keyframes to represent each shot. The keyframes are extracted as a by-product of multi-

resolution analysis for shot boundary detection. We extend the concept of finding the scene transitions as local maxima's in the feature space; the local minima's can be chosen to represent the keyframe. Only the Resolution 3 (low resolution) is used to find the local minima points. For every shot two minima's are identifies, one in the DCT DC feature space and the other in the MA64 feature space. The color histogram distance between these two representative frames are computed and thresholded to chose either one or both the frames. Also a time constraint of 1 sec distance between the two keyframes is imposed. This ensures that the keyframes are distant and well represents the action in the video. The DCT DC selection results in color constant keyframes, where as the MA64 minima represents minimal motion angle change in the consecutive frames.

4. Experimental Results

The effectiveness of the algorithm was evaluated on TREC-2002 test data set. We submitted 2 runs represented as Nus1 and Nus2. The video contained a total of 2090 transitions. About 70% of them were cuts and 30 % gradual transitions and others. The results suffer from a poor detection of gradual transitions. We observed that our system throws many short gradual transitions (SGT) for single long gradual transitions. Also Fade-In-Out was considered as two separate transitions. We never eliminated the start and end transitions. By changing the SGT value to 4 we see that there is a 7% improvement in GT recall and 5% improvement in GT precision. Also we made small experiments to merge neighboring SGT's. The results show a 7% improvement in GT recall and 15 % improvement in Frame recall. These results are tabulated in the following table.

System Description	**All**		**Cuts**		**Gradual**			
	Rec	*Prec*	*Rec*	*Prec*	*Rec*	*Prec*	*F-Rec*	*F-Prec*
Nus1	0.621	0.615	0.742	0.670	0.313	0.411	0.301	0.833
Nus2	0.594	0.614	0.707	0.693	0.306	0.369	0.331	0.848
Nus1(with SGT=4)*	0.63	0.625	0.732	0.692	0.374	0.42	0.268	0.838
Nus1(With Merge)*	0.6	0.675	0.685	0.75	0.382	0.465	0.455	0.654

5. Conclusion

In conclusion, it has been demonstrated that TMRA framework offers a general and novel approach to flexibly and accurately probe the structure and content of digital video and meantime, it provides the ability to incorporate the new function to expand and improve the performance. Our future work is (1) to improve gradual transition detection and frame recall, (2) to investigate the active learning via artificial neural network to classify the CUT/GT, (3) to investigate the usage of other features for analyzing the video data, especially at semantic level, (4) to improve and to incorporate it into our video retrieval system.

Acknowledgements

The authors would like to acknowledge the support of the National Science and technology Board, and the Ministry of Education of Singapore for supporting this research under research grant RP3989903.

Reference

[1] Yu-Ping Wang and S.L.Lee.[1998]. "*Scale-Space Derived From B-Spline* ", PAMI Vol.20, No.10.
[2] Cohen A and Ryan R D, [1995]. *Wavelet and Multiscale Signal Processing*, Chapman and Hall Publishers.
[3] Y.Lin, M.S.Kankanhalli, and T.S.Chua,[2000] "*Temporal Multi-resolution Analysis for video Segmentation*", Proc. SPIE Conf. Storage and Retrieval for Media Database VIII, SPIE Vol. 3970, SPIE Press, Bellingham, Wash., Jan. 2000, pp.494-505.