AN UNIFIED FRAMEWORK FOR SHOT BOUNDARY DETECTION VIA ACTIVE LEARNING

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

ABSTRACT

Video shot boundary detection 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 paper, we expanded our previous temporal multi-resolution analysis (TMRA) work by introducing the new feature vector based on motion. Further we employ the support vector machine (SVM) to refine the classification of shot boundaries. The resulting framework has been tested on the MPEG 7 video data set, and has been shown to have good accuracy for both the detection of abrupt and gradual transitions as well as their boundaries. It also has good noise tolerance characteristics.

1. INTRODUCTION

The rapid accumulation of huge amount of digital video data in archives has led to many video applications. These applications should have the ability to represent, index, store and retrieve video efficiently. Since video is a time-based media, it is very important to break the video streams into basic temporal units called shots [1]. The temporal partitioning of video is generally called video segmentation or shot boundary detection [1][3][5]. To fulfill the task of partitioning the video, video segmentation needs to detect the joining of two shots in the video stream and locate the position of these joins. These joins appear to be of two different types, abrupt transition (CUT) and gradual transition (GT), based on the techniques used in the editing process [2].

Due to the presence of these types of transitions and the wide varying lengths of GTs, the task of detecting the type and location of transitions in video is very complex. In fact, the detection of the transitions of video is a temporal multi-resolution problem. Information across resolutions will 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 [7], we employ Canny-like B-Spline wavelets in this multi-resolution analysis. The resulting system, based on color feature, provides a general framework to detect both CUTs and GTs effectively [8]. However, even though the system shows very high recall rates, it suffers from poor precision. This is because the system is sensitive to luminance/motion noises and threshold selection. The other major problem is the flash and camera/object motion, resulting in many falsely detected transitions.

In this paper, we extend our previous TRMA system [8] to incorporate new motion-based feature vector and use SVM to improve the accuracy of the classification of shot transitions. Tests show that the resulting system is able to improve the precision of shot boundary detection while retaining high recall. This paper discusses the above extensions.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 describes our previous TMRA framework, while Sections 4 and 5 describe the extensions. Section 6 discusses the experimental results, and Section 7 presents the conclusion and outlines of future work.

2. RELATED WORK

Much work has been done on detecting the CUTs and GTs. Zhang et al. [3] developed one of the most successful early methods, called twin-comparison method, which detects CUTs and GTs by applying different thresholds based on differences of color histograms between successive frames. They extended their method to work directly on the compressed domain by comparing DCT coefficients [6]. These, together with most existing methods, suffer a lot from threshold selection and noise [1].

There are other approaches to tackle this problem. Hampapur et al. [2] proposed the model-based method by studying the video production techniques, utilizing different models for different editing effects. Yu & Wolf [5] used wavelet to decompose every frame into low-resolution and high-resolution components. They extracted edge spectrum average feature in the high-resolution component to detect fade, and applied double chromatic difference on the low-resolution component to identify the dissolve transitions.

Our approach differs from these previous works in the following ways. First, we measure the difference between the sets of frames instead of only on every two successive frames. Second, we perform multi-resolution analysis on the temporal domain instead of only in the spatial domain as is done in most existing multi-resolution methods. Third, in our previous work [8], our system suffered from poor precision because of the noise presence in the multi-resolution wavelet coefficients, and there was no means to automatically select a threshold value to eliminate this noise. Our active TMRA method overcomes the above constraints.

3. THE TMRA FRAMEWORK

This section discusses the existing temporal multi-resolution approach (TRMA) to detect shot boundaries.

3.1 Video Representation

We model 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 a video can be modeled in N-dimensional spaces of different features. For example, we can choose DC64 color histogram as our feature space. The DC64 color histogram is computed by extracting the DCT DC value for each macroblock in the video frame. The value can be quantized levels, say, 64 values to form the DC64 color histogram.

By empirically observing GTs, we find that different types of GTs exist like fade in/fade out, dissolve, wipe and 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 all the characteristics of all kinds of shot transitions efficiently. It is observed that different types of shot transitions are observable at different resolutions in the feature space. For example, we could see CUTs both in a fine resolution (between two successive frames) and a coarse observation (across several frames), while GT only shows up clearly as a transition in a coarse resolution. So the transition must be defined with respect to different resolutions. By viewing the video at multiple temporal resolutions, the detection of CUTs and GTs can be unified. Figure 1 shows a multi-resolution analysis map based on DC64 feature for a video stream. The Figure clearly shows the locations of the CUT and GT. Note that the transition corresponds to GT only shows up as a local maxima at low-resolution scale of 4.

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.

3.2 Applying Wavelet

Wavelet provides a good mathematical basis for video analysis. In the analysis, we need to construct a scale space. The Gaussian scale-space approach is widely adopted. This is because the Gaussian function is the unique kernel which satisfies the causality property as guaranteed by the scaling theorem. It states that no new feature points are created with increasing scale [4]. 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 correspond to the local maxima of the wavelet transform. Thus a maxima detection of the wavelet transform is equivalent to boundary detection. If the mother wavelet 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 s 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. The detailed derivation of Equations (1-4) can be found in [8]. The local maxima of the resulting signal will indicate where the transitions happen, and the magnitude of the maxima will show the strength of the transitions. Since the analysis of video involves the processing of huge amount of data, the choice of a suitable kernel that facilitates fast processing is important. To achieve this, we select the Canny-like B-spline wavelets since they have fast algorithm independent of the resolution, and carry the good features of Canny [4]. Figure 1 shows the multi-resolution coefficients after the transformation for the DC64 feature space.

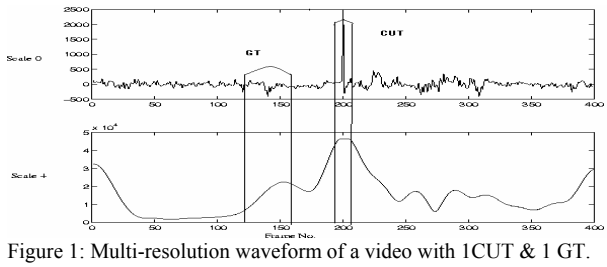

Figure 1: Multi-resolution waveform of a video with 1CUT & 1 GT.

4. MODIFIED TMRA

This section discusses the extension to the TMRA framework, by incorporating new motion-based feature and employing adaptive thresholding to make the system more dynamic.

4.1 Motion-Based Feature Vector

The previous TRMA approach [8] employed only the DC64 color histogram features. Large-scale tests showed that the TRMA method outperformed the Twin-Comparison method, and was able to achieve an F_1 value of 0.86 and 0.77 for CUT and GT detection respectively. The test, however, also showed that the use of only color histogram-based features has difficulties in improving both the recall and precision of the shot boundary detection at the same time. In addition, it has difficulty in locating precise boundary of GT due to the flash and camera/object motions. To overcome this problem we construct a motion-based feature using the motion-vectors of MPEG compressed stream.

The new feature is called the MA64 direction histogram. It is computed using the motion vectors for each macroblock 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 counts of forward predicted, backward predicted, intra and skip macroblocks.

Both the DC64 and MA64 features are used in the multi-resolution wavelet analysis framework to detect shot transitions.

4.2 Adaptive Thresholding

The problem of choosing the appropriate thresholds is a key issue to all shot boundary detection methods, including TRMA. Heuristically chosen global threshold is not suitable as the shot

content changes from scene to scene. 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 be adaptively adjusted based on the sliding window size and the standard deviation of DC64 feature of the neighborhood frames. For different video clips, their standard deviations (STD) are different. Also, the choice of sliding window size is also very important. We choose the sliding window size as the sum of max interval of peak points and max interval of valley points:

$$size = dist_p + dist_v, \quad with$$
$$dist_p = \arg\max_{i \in N_p} \{D_i^{(p)}\}, dist_v = \arg\max_{i \in N_v} \{D_i^{(v)}\} \tag{5}$$

Where N_p and N_v respectively denote the number of the peak and valley points; $D_i^{(p)}$ and $D_i^{(v)}$ give the interval of the neighborhood peak and valley points.

At each sliding window position, we calculate the average magnitude of the feature point and use that as the dynamic threshold for that point. Figure 2 shows the adaptive threshold for the wavelet coefficients. Our tests show that the adaptive threshold removes most of the noise peak points due to brightness/contrast variations, blurring and small motions.

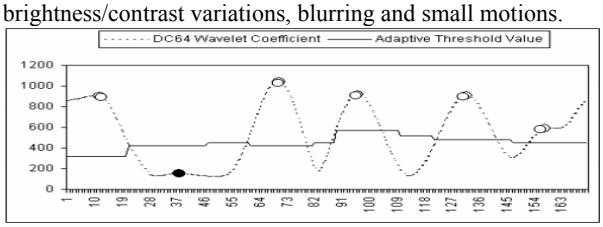

Figure 2: Adaptive threshold for wavelet coefficients

4.3 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. Here we analyze both the color and motion characteristics of the video stream using the DC64 and MA64 features simultaneously. The idea is to use the low resolution wavelet coefficients (we use the DC64 feature at resolution 3) to help in detecting the occurrence of the transition, while examining the high resolution information (we use MA64 feature at resolution 0) to locate the start and end of the transition.

To locate potential transitions, we start with resolution 3 (the 4th scale) of the wavelet analysis as at this resolution, most CUTs as well as GTs would show up as local maxima. The use of resolution 3 also ensures that no GT is lost in the analysis. In general, not only the true transitions, but many noise and object motions will also show up as local maxima in the multi-resolution wavelet analysis. As wavelet transform is a smoothing kind of function, part of the noise would have already been removed in the lower resolutions. Also, most noise would be severely degraded as we move from high to low resolution space. Thus by tracing the local maxima from the high to a low resolution scale in the DC64 feature space, we can eliminate most of the maxima that correspond to noise. However, many maxima correspond to fast camera/object motions will still remain.

After we have identified the local maxima points at the 3rd lower resolution, we use these local maxima points as the anchor points. We trace up to the higher resolution of the motion-based MC64 wavelet coefficients. As the motion vector is sensitive to changes in scene contents, we expect the beginning and end of the GTs to show up as clear local maxima in the MA64 feature space at resolution 0. They thus provide the basis to locate the transition boundaries precisely as illustrated in Figure 3.

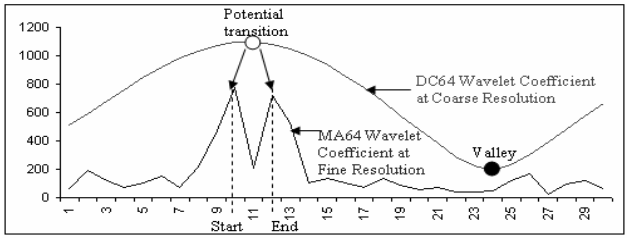

Figure 3: Locating transition boundaries using multi-resolution

5. ATMRA

The use of modified TRMA has improved the precision of the results by about 6% over the TRMA system (see results in Table 1). However, we found that there are still too many false transitions. This is because of noise as well as various kinds of object and camera motions, which are hard to be captured by using generic heuristic methods as is done previously. Hence, to further filter out the noise across all types of video content, we introduce an additional elimination-verification step based on machine learning. This active learning based system (ATMRA) is based on SVM. This section discusses the overall ATMRA system and the features to be extracted for SVM training.

5.1 Feature Selection

In order to accurately classify the potential transitions into the types of CUT and GT, and to filter out those due to noise and motions, we need to further characterize the transitions. As we expect the changes during a gradual transition to be relatively smooth, in principle, we expect the mean absolute differences (MAD) of DC64 and MA64 for the "true" transitions (CUT/GT) to be consistent. In other words, at the "true" CUT and GT points, we expect the MADs to be consistent across both DC64 and MA64 feature space. If the MAD changes are not consistent across DC64 and MA64, it implies that it is a wrong transition caused by noise or camera/object motion.

Based on this analysis, we derive the following features to capture the consistency among the features during the transitions. First, we derive four kinds of quadratic differences to measure the various distances between the mean absolute differences of feature vectors at each potential transition. They are represented as -- $QMAD_{before}$ (similarity before the transition), $QMAD_{inter}$ (similarity within the transition), $QMAD_{after}$ (similarity after the transition) and $QMAD_{ba}$ (similarity before and after the transition):

$$QMAD_{before} = \left(\sum_{i=start-k}^{start-1} \sum_{j=i+1}^{start} | f_i - f_j | \right) / C_k^2 \tag{6}$$

$$QMAD_{inter} = \left(\sum_{i=start}^{end-1} \sum_{j=i+1}^{end} | f_i - f_j | \right) / C_{end-start+1}^2 \tag{7}$$

$$QMAD_{after} = \left(\sum_{i=end}^{end+k-1} \sum_{j=i+1}^{end+k} | f_i - f_j | \right) / C_k^2 \tag{8}$$

$$QMAD_{ba} = \left(\sum_{\substack{i=start-k \\ i \notin (start, end)}}^{start-1} \sum_{\substack{j=i+1 \\ j \notin (start, end)}}^{end+k} | f_i - f_j | \right) / C_{2k}^2 \tag{9}$$

Where *start* and *end* represents the beginning and ending frame number of the potential transition. k represents the computing range ($2 \le k \le 9$), f_i denotes the feature, which in this case, the DC64 and MA64 features. C_k^r is the normalization factor.

Next we compute the ratios of QMADs defined as:

$$ratio1 = QMAD_{inter} / \max(QMAD_{before}, QMAD_{after}) \quad (10)$$

$$ratio2 = QMAD_{inter} / QMAD_{ba} \quad (11)$$

The ratios are used to better differentiate between different types of transitions. Finally, we also compute the counts of intra and skip macroblocks. This is used to capture the motion information within the transitions.

The above set of features is then used to train the SVM model.

5.2 Support Vector Machine (SVM)

Given the set of training data consisting of both positive and negative examples, we derive the above set of features for each transition. We train an SVM-based classifier to classify the remaining potential transitions into three classes of CUTs, GTs and false transitions.

We collected a large variety of training data in different genres (home video, animation, news etc.) from the MPEG-7 dataset. The dataset consists of about 5,000 transitions (positive and negative). We randomly selected 20~30 % of dataset as the training data and randomly chose 15~20% of remaining dataset as the validating data. We used SVM model for multi-class classification. Currently, we only classified a transition into the types of: CUT, GT or False. Of course, we can further classify the GT into other fine types such as dissolve, fade in/out, morph etc.

After training and validating the SVM model, we use this SVM model to classify the potential transitions and eliminate false transitions caused by flash and camera/object motions.

6. EXPERIMENTAL RESULTS

The effectiveness of the algorithm was evaluated on the MPEG - 7 test dataset using the precision, recall and F_1 measures, which are widely used in the field of information retrieval. There is a total of about 13 hours of video, which contains about 5,256 CUTs and 1,085 GTs.

For comparison purpose, we tuned a Twin-Comparison method (Twin-Comp) [3] by selecting the best possible thresholds, and used that to provide the baseline performance. We compared the results of the Twin-comparison method with: (a) the original TRMA method [8]; (b) the modified TRMA (M-TRMA) as described in Section 4; and (c) ATRAM described in Section 5. Table 1 summarizes the results.

Table 1. Comparison of results

	CUT			GT		
	Pr.	Re.	F_1	Pr.	Re.	F_1
Twin-Comp Mth	0.40	0.92	0.56	0.15	0.58	0.24
TRMA	0.77	0.98	0.86	0.65	0.97	0.77
MTRMA	0.88	0.95	0.91	0.77	0.91	0.83
ATRMA	0.96	0.93	0.94	0.88	0.89	0.88

From the table, we can see that the basic TRMA method outperforms the Twin-Comparison method by a large margin, especially for gradual transitions. The use of motion features in MTRMA improves the performance of TRMA by 4% and 5% in F_1 measure for CUT and GT respectively. Finally, the use of active learning based on SVM (ATRMA) further improves the performance over TRMA by 8% and 11% in F_1 measure for CUT and GT respectively.

We have participated in this year's TREC [9] and tested the system on the TREC-2002 Shot boundary detection corpus containing approximately 2,000 transitions. The tests again demonstrate that ATRAM out-performs the original TRMA method by over 10% in F_1 value.

7. CONCLUSION AND FUTURE WORK

In this paper, it is shown that a temporal video sequence can be modeled as a trajectory of points in the multi-dimensional feature space. By studying different types of transitions in different resolutions, it is observed that the shot boundary detection is a temporal multi-resolution phenomenon. The tests show that the ATMRA framework offers a general and novel approach to flexibly and accurately probe the structure and content of digital video. It results in very high F_1 performance for both the CUT and GT detections. Our future work includes: (a) to improve the gradual transition detection and frame recall; and, (b) to investigate the use of other features to analyze the video data, especially at the semantic level.

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.

REFERENCES

[1] F. Idris and S. Panchanathan [1997]. Review of Image and Video Indexing Techniques, Journal of Visual Communication and Image Representation, 8(2), June, 146-166.

[2] A. Hampapur, R. Jain and T. E. Weymouth [1994]. Digital Video Segmentation, ACM Multimedia Conference.

[3] HongJiang Zhang, Atreyi Kankanhalli, Stephen W.Smoliar, [1993]. Automatic partitioning of full-motion video, ACM Multimedia Systems, 1 (1), 10-28.

[4] Yu-Ping Wang and S.L. Lee.[1998]. Scale-Space Derived From B-Spline, PAMI 20(10).

[5] Hong Heather Yu, Wayne Wolf [1998]. Multi-resolution video segmentation using wavelet transformation, Storage and retrieval for Image and Video Database (SPIE), 176-187.

[6] HongJiang Zhang, Chien Yong Low, Yihong Gong and Stephen W.Smoliar,[1995]. Video Parsing Using Compressed Data, Multimedia Tools and Applications, 1(1), 91-111.

[7] A Cohen A & R.D. Ryan [1995]. Wavelet and Multiscale Signal Processing, Chapman and Hall Publishers.

[8] 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, 94-505.

[9] TREC-2002 Video Track: Shot Boundary (SB) Measures at http://www-nlpir.nist.gov/projects/t2002v/sbmeasures.html