Motion Estimation of Projected Spatio-Temporal Slice for Shot Boundary Detection

Qingxiu Du, Haibo Zheng, Shuwu Zhang
Institute of Automation, Chinese Academy of Sciences
Beijing, China
qingxiu.du@ia.ac.cn haibozheng@hotmail.com swzhang@hitic.ia.ac.cn

Abstract—In most of the existing shot boundary detection algorithms, the false/miss detection problem caused by motion is very serious. In this paper, firstly, we propose a new spatio-temporal slice called projected spatio-temporal slice (PSTS) that can effectively eliminate disturbance caused by motion. Then we present approaches for detecting camera cuts, fades and dissolves based on motion estimation of PSTS. Experimental results indicate that no matter under what kind of motion, it can detect shot transitions fast and effectively, while high recall and precision rate achieved.

Keywords-projected spatio-temporal slice, STS, PSTS, motion estimation, cuts, fades, dissolves, shot boundary detection

I. INTRODUCTION

Nowadays, advances in multimedia technology coupled with explosive growth of internet and the availability of high computing resources at affordable cost have led to the wide-spread use of digital video for varied application. The increased availability and usage of digital video has created the need for automatic video content analysis techniques. However, it is well known that we still have a limited number of tools for describing, organizing, and managing video data [1]. In order to efficiently manage a digital video database and execute fast video indexing/retrieval, video shot boundary detection must be performed.

There are a number of different types of transitions or boundaries between shots. However, 99% of all these types fall into hard cuts, dissolves, fades [2]. Therefore, many researchers only focus on the detection of cuts, fades and dissolves. In the past decade, hundreds of shot change detection algorithms have been proposed. In general, we can categorize them as basic comparison method, clustering-based segmentation, and compressed domain methods.

Basic comparison methods can be subdivided into pixel difference, histogram methods and statistical difference. Pixel difference methods are very sensitive to noise and camera motion. Histogram-based image comparison methods are quite robust against image noise and object movements. There are few drawbacks though. First, the histogram describes only the distribution of color or gray scale values. It does not contain any spatial information of the image. Another drawback of histograms is that small image regions, which may still considered visually significant, may not produce any strong peaks in histograms and would therefore be neglected in image comparison. Also, a simple histogram distance measure, analogous to human visual perception system, is hard to define. Statistical difference methods are based on pixel differences, but instead of s simple sum of absolute differences, more descriptive statistics are calculated.

The performance of the basic comparison methods rely on suitable thresholding of similarities between successive frames. However, the thresholds are typically highly sensitive to the type of input video. The drawback is overcome by the application of unsupervised clustering algorithm. The main advantage of the clustering-based segmentation is that it is a generic technique that not only eliminates the need for threshold setting but also allows multiple features to be simultaneously to improve the performance.

DCT-based methods and MPEG-based methods are the main methods of the compressed domain methods. Compared with uncompressed domain method, it are efficient than techniques in uncompressed domain. These techniques normally use the encoded features of video, such as in MPEG videos, information about motion vectors, macro block types, and DCT coefficients can be used. Discarding the decompression task reduces the processing time; however this requires a number of adaptive algorithms for every compressed type.

In this paper, we propose a new spatio-temporal slice called projected spatio-temporal slice (PSTS) that can effectively eliminate disturbance caused by motion, then we present approaches for detecting camera cuts, fades, and dissolves based on motion estimation of PSTS as well as a 3-pass detection strategy.

The rest of this paper is organized as follows. First, PSTS is introduced in Section 2. Section 3 describes the proposed shot boundary detection method. The experimental results and analysis are presented in Section 4. Finally, conclusions and future works are given in Section 5.

This work has been supported by National Natural Science Foundation of PRC (No. 60773038), National 863 Program (No. 2006AA01Z130) and National Key Technology R&D Program (No. 2008BAH21B03-04 and No. 2008BAH26B03).

978-1-4244-4657-5/09/$25.00 ©2009 IEEE

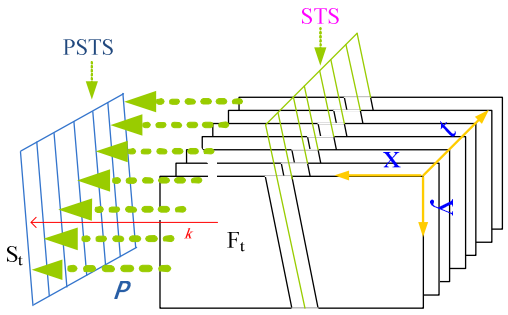


Figure 1. STS and PSTS

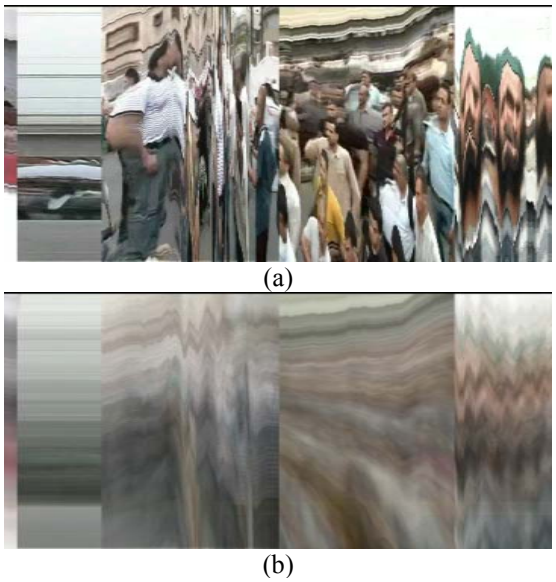
II. PROJECTED SPATIO-TEMPORAL SLICE

Traditionally, spatio-temporal slice (STS) is built by selecting one segment of pixels from each frame and concatenating them to a 2D image (shown in Fig. 1). The concept was initially proposed by Edward H Adelson and James R Bergen[5].

In this paper, a novel slice called *projected spatio-temporal slice* (PSTS) is proposed as shown in Fig. 1, S_t is a line which is the projection of the corresponding frame F_t along direction k, and PSTS is built by concatenating S_t on time order to form a 2D image. Although P can be oblique projection or orthogonal projection, orthogonal projection is frequently adopted in practice for convenience.

If orthogonal projection along horizontal direction is used, pixel $S_t(y)$ in PSTS can be mathematical expressed as

$$S_t(y) = \sum_x F_t(x, y) \tag{1}$$



(a)

(b)

Figure 2. STS(a) and PSTS(b) of a same video segment with motion

As can be seen from Fig. 2, STS and PSTS all contains a huge quantity of information about objects motion, camera zooms and motions including tilts and pans, and are all split into two homogeneous zones by shot transitions in the direction of the time axis. This is due to the disappearance of scene objects/zones of the current shot and appearance of the components of the next. Therefore, to detect the shot transition, we can analyze the continuity of the STS or PSTS signal along time axis.

As STS is extracted from one position, the content of STS is chaos when the pixels cross the extract position changed and conducts more miss/false hits. But in PSTS, as each column of PSTS is projection result of one frame, the content of PSTS is more homogeneous than especially in case of motion.

III. THE ALGORITHM

A. Normalization

Before apply our algorithm, normalization should be performed firstly. In this paper, PSTS is normalized to [0, 255].

B. Motion Estimation

Each pixel $S_t(y)$ in PSTS, its position must be one of three cases:

1) *At beginning of the shot,*
2) *At middle of the shot,*
3) *At end of the shot.*

Here, a short example of motion estimation of pixel $S_t(y)$ in PSTS is given (shown in Fig. 3). L and R is the best match point of $S_t(y)$ in the left and right neighborhood respectively. The left-right differences ratio of $S_t(y)$ is defined as

$$r_{t,y} = \frac{L_{t,y}}{R_{t,y}} \tag{2}$$

$$L_{t,y} = \min_{k\in[t-w-1,t-1],l\in[y-\frac{h}{2},y+\frac{h}{2}]} \left(\sum_{m\in\{R,G,B\}} | S_k(l) - S_t(y) | \right) \tag{3}$$

$$R_{t,y} = \min_{k\in[t+1,t+w+1],l\in[y-\frac{h}{2},y+\frac{h}{2}]} \left(\sum_{m\in\{R,G,B\}} | S_k(l) - S_t(y) | \right) \tag{4}$$

In which $L_{t,y}$ means difference between $S_t(y)$ and L, $R_{t,y}$ means difference between $S_t(y)$ and R.

And we can get the range of possible values of $r_{t,y}$ in the three cases mentioned above.

$$r_{t,y}\begin{cases} <<1 & in\ \ case\ \ 1) \\ \approx 1 & in\ \ case\ \ 2) \\ >>1 & in\ \ case\ \ 3) \end{cases} \tag{5}$$

As one column of PSTS in correspondence with one frame, the feature of frame can be expressed with the left-right differences ratio of column of PSTS. The left-right differences ratio of column of PSTS can be defined as

$$D(t)=\sum_{y} r_{t,y} \tag{6}$$

C. Second-Max Ratio Criterion

Although $D(t)$ can tolerant toward motion, $D(t)$ still change greatly due to the type, size and quality of videos. We use the second-max ratio criterion to deal with this situation. The second-max ratio, $R(t)$, is defined as

$$R(t)=\frac{D(t)}{\max\limits_{t'\in[t-w_1,t+w_1],t'\neq t} D(t')} \tag{7}$$

D. Cut Detection

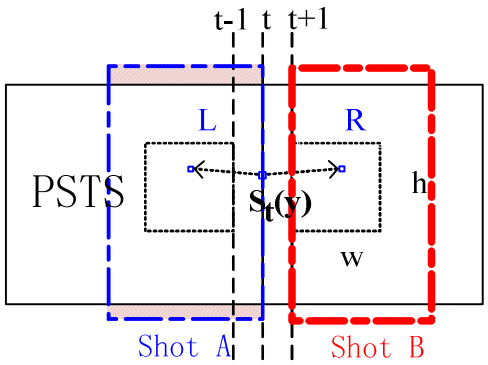


Figure 3. Motion estimation of pixel in PSTS

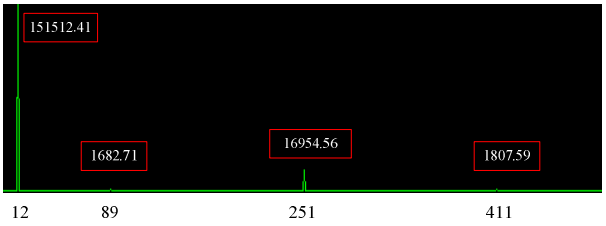


Figure 4. Example of N(t)

$$N(t)=\frac{R(t)}{R(t-1)} \tag{8}$$

In order to get a better description, we use $N(t)$ to locate the precise position which cut happened.

If current column is inside of a shot, $R(t)$ is very close to $R(t-1)$, so $N(t)$ is approximately equal to 1. If current column is the first frame of a shot, $R(t)$ is very large, and $R(t-1)$ is very small, so $N(t)$ is a very large value. If current column is the last frame of a shot, $R(t)$ is very small, and $R(t-1)$ is very large, so $N(t)$ is a very small value.

Fig. 4 gives out an example of $N(t)$ which is the result of PSTS shown in Fig. 2(b). The $N(t)$ corresponding with the real cut is very large, and others are all below or near around 1 whether there is motion or not.

1) Threshold-independent filter

By analyzing equation 5, 6 and 7, we can know when a cut happens, $D(t)$, $R(t)$ and $N(t)$ should satisfy the following conditions

a) $D(t)$, $D(t-1)$ *is the maximum and the minimum in neighborhood, respectively.*

b) $R(t)>1$.

c) $N(t)>1$ *and* $N(t)$ *is the maximum in neighborhood.*

According to the above 3 conditions, a set of possible cut transitions can be obtained. For each element in the possible set, the corresponding $N(t)$ is used to indicate the possibility whether this is a true cut or not.

For slice j, the possible set of cut and the corresponding score set can be defined as

$$Maybe_j=\{t_{kj}\} \tag{9}$$

$$Score_j=\{N(t_{kj})\} \tag{10}$$

In which j is the index of PSTS, t_{kj} is the position of possible cut.

2) Cross-validation

For the same video segment, if we select different project direction, we can get N different PSTSs. Since cut is a dramatic globe change in the content, so each PSTS will show the boundary clearly in the same position. According to the characteristics of this, we use multiple PSTSs to do cross validation for results after threshold-independent filter.

Suppose we have PSTSs, we only select the points appeared in all possible set as candidates.

$$Maybe = \{t_k\} \tag{11}$$

$$Score = \{N^{'}(t_k)\} \tag{12}$$

$$t_k \in Maybe_j, 0 \le j < N \tag{13}$$

$$N^{'}(t_k) = \frac{\sum_{j=0}^{N-1} N(t_{kj})}{N} \tag{14}$$

3) 3-Pass check strategy

After cross-validation, we have a Maybe set and Score set. As we can see from the Fig. 5, score for each element in candidates corresponding to the confidence level which indicates whether it is a cut.

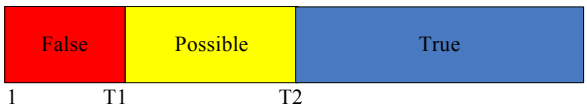


Figure 5. Score region and confidence level

Now, we give the 3-pass check strategy in details.

a) If score of the check point is great than T2 (falls into the true region), we mark the corresponding point as cut.

b) If score of the check point is less than T2 and great than T1 (falls into possible region), we use [6] to test whether it is a cut.

c) If score of the check point is less than T1 (falls into the false region), we simply filter out the corresponding point.

Although method used in [6] is very time exhausting, but fortunately, the points fall into the possible region is little.

E. Monochrome Frame Detection

As PSTS can represent continues spatial and time information, monochrome frame can be easily detected. For convenience, some symbols are defined.

$$B_t(y) = \begin{cases} 1 & M_t(y) - m_t(y) < T_{b1} \\ 0 & otherwise \end{cases} \tag{15}$$

$$M_t(y) = \max_{m \in \{R,G,B\}} S_t(y)[m] \tag{16}$$

$$m_t(y) = \min_{m \in \{R,G,B\}} S_t(y)[m] \tag{17}$$

$$B_t = \frac{\sum_y B_t(y)}{H} \tag{18}$$

$$M_t = \max_y M_t(y) \tag{19}$$

$$m_t = \min_y m_t(y) \tag{20}$$

In which H is the height of PSTS.

A frame is declared as a monochrome frame if the corresponding column of PSTS satisfies the followings 2 conditions.

1) $B_t > T_{b2}$

2) $B_t < T_{b3}$ *or* $m_t > T_{b4}$

In which $T_{b1}, T_{b2}, T_{b3}, T_{b4}$ are practical threshold.

F. Dissolve Detection

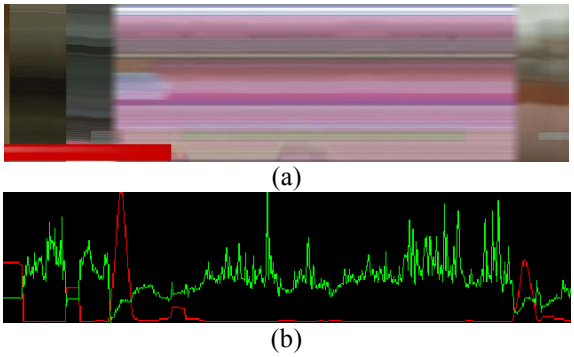
We propose a PSTS-based method for dissolve detection which combines the skipping column difference and the degree of linearity of a sequence of columns. Suppose we currently have a sequence of w images which is a segment in the video sequence start with S_{t-w-1} and ending with S_t. We define the current skipping difference as

$$J(t) = \| S_{t-w-1} - S_t \| = \sum_j \left(\min_{k \in [j-w_2, j+w_2]} \left(\sum_{m \in \{R,G,B\}} | S_{t-w-1}(k)[m] - S_t(j)[m] | \right) \right) \tag{21}$$

We define the current normalized linear error along this part of video as

$$LE(t) = \frac{\sum_{i=0}^{w-1} \| S_t - ((1 - \frac{i}{w-1}) S_{t-1} + \frac{i}{w-1} S_{t-w-1} \|}{J(t)} \tag{22}$$

We detect a dissolve by presence of a peak of J(t), and the corresponding start and end point of the linear increase part of LE(t) is the start frame and end frame of a dissolve transition.



(a)

(b)

Figure 6. PSTS(a) and corresponding J(t)(b)(red), LE(t)(b)(green)

IV. EXPERIMENTAL RESULTS

In our implementation, we use two PSTSs obtained by project along horizontal and vertical direction for cross validation. In 3-pass check strategy, T_1, T_2 is 10 and 200, respectively. In monochrome frame detection, $T_{b1}, T_{b2}, T_{b3}, T_{b4}$ is 6, 0.85, 70 and 180, respectively.

To measure the efficiency of the proposed approach, we use the classical definition of recall, precision and F1.

$$V = \frac{C}{C + M} \tag{23}$$

$$P = \frac{C}{C + F} \tag{24}$$

$$F1 = \frac{2 \times P \times V}{P + V} \tag{25}$$

In which the symbols stand for: C, the number of correctly detected ("correct hits"), M, the number of not detected ("missed hits") and F, the number of falsely detected ("false hits"). $F1$ is a combined measure that results in high value if, and only if, both recall V and precision P result in high values.

To test the robust stability of our algorithm, a variety of video were utilized including MTV, movies, news and sports. They are typical in size, motion intensity and quality.

From the analysis of the algorithm, we know that motion estimation and the validation [6] of the possible region occupy most of the total complexity. Table 2 gives the distribution of three confidence levels, we can see that the thresholds of our algorithm are robust to various types of video. For points in possible region, we apply method used in [6] to validate where it is a true shot transition or not. Due to the amount of points in possible region is very small, so the complexity is low, relatively.

Table 3 gives the results for each type of shot transition detection. The total recall and precision are all obtained with high value. According to the results obtained, the algorithm is characterized by high precision and recall. Gradual transition detection is relative poor, because the shape detection implement is not as good as we expected.

TABLE I. TEST SET USED IN EXPERIMENT

Runid	Type	Duration(secs)	Cuts	Fades+Dissolves
R0	MTV	450	146	47
R1	Movie	1705	531	18
R2	News	1800	450	21
R3	Sports	1445	177	25

TABLE II. DISTRIBUTION OF SCORES

Runid	False region	Possible region	True region
R0	0.626	0.066	0.308
R1	0.522	0.045	0.433
R2	0.687	0.022	0.291
R3	0.649	0.054	0.297

TABLE III. THE RESULTS OF SHOT BOUNDARY DETECTION

Runid	Total			Cuts			Fades+Dissolves		
	V	*P*	*F1*	*V*	*P*	*F1*	*V*	*P*	*F1*
R0	90	100	94.7	97	100	98.5	55	100	71
R1	94.2	98.3	96.2	95.7	99.4	97.5	57.1	66.7	61.5
R2	96.7	97.1	96.9	100	97.7	98.8	23.8	62.5	34.5
R3	82.7	93.3	87.7	91.5	98.8	95.0	20.0	33.3	25.0

V. CONCLUSION

In this paper, we propose a new PSTS, and present a new method to detect shot boundary. Experimental results show that our algorithm has good performance for speed, recall rate and precise rate even in case of large movement.

REFERENCES

[1] N. Dimitrova, H. J. Zhang, B. Shahraray, I. Sezan, T. Huang, and A. Zakhor, "Applications of video-content analysis and retrieval," IEEE Multimedia, vol. 4, no. 3, pp. 42–55, Jul./Sep. 2002.

[2] R. Lienhart, "Comparison of Automatic Shot Boundary Detection Algorithms," SPIE Storage and Retrieval for Still Image and Video Databases, pp. 3656:290-301, 1999.

[3] Chih-Wen Su, Hong-Yuan Mark Liao, etc., "A Motion-Tolerant Dissolve Detection Algorithm," IEEE Transactions on multimedia, Volume 7, Number 6, pp. 1106-1113, December 2005.

[4] Edward H. Adelson, J.R.B., "Spatiotemporal energy models for the perception of motion," Journal of optical America, 1985. 2 No 2: pp.284-299.

[5] Aissa Saoudi, and Hassane Essafi, "Spatio-Temporal Video Slice Edges Analysis for Shot Transition Detection and Classification.," International journal of signal processing, Volume 4, Number 1, pp. 189-194, 2007.

[6] Ming Luo, Daniel DeMenthon, David Doermann, "Shot Boundary Detection using Pixel-to-Neighbor Image Differences in Video," TRECVID Workshop 2004.