UDC 004.421
Original Scientific Paper

Shot Boundary Detection in Soccer Video using Twin-comparison Algorithm and Dominant Color Region

Matko Šarić *msaric@fesb.hr*
University of Split
FESB Split

Hrvoje Dujmić *hdujmic@fesb.hr*
University of Split
FESB Split

Domagoj Baričević *dbaricev@fesb.hr*
University of Split
FESB Split

Abstract

The first step in generic video processing is temporal segmentation, i.e. shot boundary detection. Camera shot transitions can be either abrupt (e.g. cuts) or gradual (e.g. fades, dissolves, wipes). Sports video is one of the most challenging domains for robust shot boundary detection. We proposed a shot boundary detection algorithm for soccer video based on the twin-comparison method and the absolute difference between frames in their ratios of dominant colored pixels to total number of pixels. With this approach the detection of gradual transitions is improved by decreasing the number of false positives caused by some camera operations. We also compared performances of our algorithm and standard twin-comparison methods.

Keywords: Video segmentation, cut detection, gradual transition detection, dominant color region, twin-comparison algorithm

1. Introduction

The use of video data in the multimedia environment is increasing rapidly, and tools to handle large volumes of video data are required. Temporal video segmentation is the first step towards automatic annotation of digital video sequences. Its goal is to partition video into a set of meaningful and manageable segments (shots).

A shot is defined as an unbroken sequence of frames taken from one camera. There are two basic types of shot transitions: abrupt and gradual. A comprehensive survey of shot detection algorithms is given in [1]. The simplest method for cut detection is to calculate the absolute sum of pixel differences and compare it against a threshold [2]. The problem with this method is its sensitivity to camera and object movements. A better method is to compare block regions instead of individual pixels. Sensitivity to camera and object movements can be further reduced by comparing histograms of successive images. The idea behind histogram based approaches is that two frames with the same background and the same object (although moving) will have little difference in their histograms. The techniques mentioned so far are single threshold based approaches for cut detection and they are not suitable to detect gradual transitions. A simple and effective two threshold approach for gradual transition detection is the twin-comparison method [3].

These approaches for video segmentation process uncompressed video, but it is desirable to use methods that can operate directly on the encoded stream. Working in the compressed domain reduces computational complexity (there is no need for decoding/re-encoding), operations are faster because of the lower data rate of compressed video, and the encoded stream contains a set of precomputed features that can be used for temporal video segmentation. Some examples of algorithms for MPEG compressed domain are in [4] and [5]. In [10] a unified model for techniques on video-shot transition detection was introduced. This approach is based on mapping the space of inter-frame distances onto a new space of decision better suited to achieving sequence-independent thresholding.

Shot boundary detection in sports video is a challenging task. Ekin et al. [6] introduced a new feature for the temporal segmentation of soccer video - the absolute difference between two frames in their ratios of dominant (grass) colored pixels to total number of pixels. As a second feature they used difference in color histogram similarity. Our approach is based on first feature (the difference in the ratio of dominant colored pixels) and the twin-comparison algorithm.

The paper is organized as follows. In section 2 the detection of the dominant color region is described. The twin-comparison algorithm is presented in section 3. Section 4 introduces our algorithm for shot boundary detection. Finally, the results are presented in section 5.

2. Dominant color region detection

In this paper we used the method described in [6] to find the dominant color of the soccer field (a tone of green). The color image is in the RGB space with the color histogram of each component defined as H[*i*]. The dominant color is determined by the mean values of each color component which are computed around their histogram peaks. The first step is determining the peak index for each component. Next, the interval *[i_{min}, i_{max}]* is found around each peak, where i_{min} and i_{max} represent the minimum and maximum indices of the interval. These indices must satisfy equations (1)-(6). The equations define the minimum (maximum) index as the smallest (largest) index to the left (right) of the peak (including the peak) that has a predefined number of pixels (e.g. K=0.2 – 20% of peak count). The mean color is computed for each color component by (7).

$$H\left[i_{\min}\right] \geq K \cdot H\left[i_{peak}\right] \tag{1}$$

$$H\left[i_{\min}-1\right] < K * H\left[i_{peak}\right] \tag{2}$$

$$H\left[i_{\max}\right] \geq K * H\left[i_{peak}\right] \tag{3}$$

$$H\left[i_{\max}+1\right] < K * H\left[i_{peak}\right] \tag{4}$$

$$i_{\min} \leq i_{peak} \tag{5}$$

$$i_{\max} \geq i_{peak} \tag{6}$$

$$Color\,mean = \frac{\sum_{i=i_{\min}}^{i_{\max}} H[i] * i}{\sum_{i=i_{\min}}^{i_{\max}} H[i]} * Q_{size} \tag{7}$$

In (7) Q_{size} is the quantization size, and is used to convert an index to a color value. The mean of each color component is then converted from RGB to HSI. For the color pixels in each frame, we calculate the distance from each pixel to the mean color ($d_{cylindrical}$) using the robust cylindrical metric represented with equations (8)-(12):

$$d_{\mathrm{int}ensity}(j) = \left|I_j - \overline{I}\right| \tag{8}$$

$$d_{chroma}(j) = \sqrt{\left(S_j\right)^2 + \left(\overline{S}\right)^2 - 2S_j\overline{S}\cos\left(\theta(j)\right)} \tag{9}$$

$$d_{cylindrical}(j) = \sqrt{\left(d_{\text{int}ensity}(j)\right)^2 + \left(d_{chroma}(j)\right)^2} \tag{10}$$

$$\theta(j) = \begin{cases} \Delta(j) & if\ \Delta(j) <= 180^0 \\ 360^0 - \Delta(j) & otherwise \end{cases} \tag{11}$$

$$\Delta(j) = \left|\overline{Hue} - Hue_j\right| \tag{12}$$

Hue, *S* and *I* refer to hue, saturation and intensity and *j* is the *j*th pixel. Pixel *j* belongs to the dominant color region if it satisfies the constraint

$$d_{cylindrical} < T_{color} \tag{13}$$

where T_{color} is the predefined threshold which is video dependent.

3. Twin-comparison algorithm

A shot is defined as an unbroken sequence of frames taken by a camera. There are two basic types of shot transitions: abrupt and gradual. Abrupt transitions (cuts) are simpler, they occur in a single frame when stopping or starting the camera. Gradual transitions can be roughly divided into two classes: those that simultaneously but gradually affect every pixel of the image, and those that abruptly affect an evolving subset of the pixels, with the subset changing in each frame. The first class includes dissolve and fade-in/out effects. Dissolves show one image superimposed on the other as the frames of the first shot get dimmer and those of the second one get brighter. Fade out is a slow decrease in brightness resulting in a black frame; a fade in is a gradual increase in intensity starting from a black image. The second class commonly includes wipe effects. Wipe transitions are generally characterized by slowly sliding in or uncovering an image from a new shot, while simultaneously sliding out or covering up the old shot. One of the most popular methods for detecting shot transitions is comparing the histograms of consecutive frames. The assumption is that two frames which have a common background and unchanging objects will show little difference in their histograms. The basic formulation for histogram comparison is as follows: the histogram (either color or grayscale) is computed for each frame and the difference is calculated as shown in (14)

$$D(i,i+1) = \sum_{j=0}^{B-1}\left|H_i(j) - H_{i+1}(j)\right| \tag{14}$$

Where $H_i(j)$ is the j^{th} element of the histogram of the i^t frame, and B is the number of bins in the histogram. In this paper we used a difference function defined by the histogram intersection (15):

$$\begin{aligned} D(i,i+1) &= 1 - Inter\sec tion(H_i, H_{i+1}) = \\ &= 1 - \frac{1}{M}\sum_{m=1}^{M}\sum_{j=0}^{B_m-1}\min\left(H_i^m(j), H_{i+1}^m(j)\right) \end{aligned} \tag{15}$$

where M denotes the number of color components, B_m is the number of bins in the histogram of the m-th color component and H_i^m is the normalized histogram of the m-th color component.

For two identical histograms the intersection is 1 and the difference 0, while for two frames which do not share even a single pixel of the same color the difference is 1.

Histogram techniques for cut detection are based on the fact that cuts generate a big difference between frames which results in high peaks in histogram comparisons. Therefore cuts are easily detected using one threshold. Such approaches aren't suitable to detect gradual transitions. Interframe differences during gradual transitions are higher than those within a shot, but they are still significantly lower than the difference when a cut is present. Camera and object motions can cause bigger differences than gradual transitions. Hence, lowering the threshold is not a solution because it increases the number of false positives.

The twin-comparison method [3] in addition to interframe differences uses cumulative differences between frames of a gradual transition. In the first pass cuts are detected using the high threshold T_h (figure 1.) In the second pass the potential starting frame F_s of the gradual transition is detected using the lower threshold T_l. F_s is then compared to subsequent frames (figure 1). This is called an accumulated comparison because during a gradual transition this difference value increases. The end frame of the transition is detected if two constraints are satisfied: the difference between successive frames falls below T_l while the accumulated difference increases over T_h. If the consecutive difference falls below T_l before the cumulative difference reaches T_h the potential start frame F_s is dropped. In some gradual transitions the consecutive difference falls below T_l. The problem is solved by allowing a certain number of frames with low difference values before rejecting F_s. In [7] several temporal video segmentation algorithms are compared and it is found that twin-comparison is a simple algorithm that works very well.

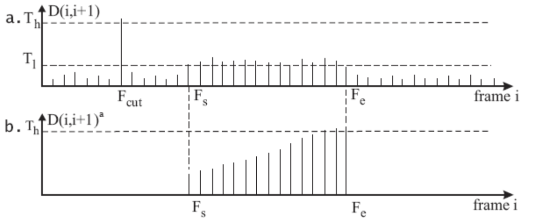

Figure 1. a) consecutive and b) accumulated histogram differences

As suggested in [9], the lower threshold T_l can be calculated using equation (16)

$$T_l = \mu + \alpha\sigma \qquad (16)$$

where μ and σ are the mean and standard deviations of interframe differences. The α is constant (5 is the suggested value). For calculating T_h the histogram of the difference values of the clip is used. After identifying the peak value, we assign T_h the index value that corresponds to half of the peak value on the right slope of the peak. T_h must be higher than mean value.

4. Shot boundary detection using twin-comparison algorithm and dominant colored pixel ratio

Sports video is one of the most challenging domains for shot boundary detection because of three observations presented in [6]. The first is a strong color correlation between sports video shots. The reason for this correlation is the existence of a single dominant color background in successive shots. Hence, shot change may not result in significant histogram difference. Second, sports video is characterized by large camera and object motions. Pans and zooms are often used, so existing shot boundary detection based on change statistics doesn't work well with sports videos. And third, in sports video clips we can find cuts and gradual transitions. Thus, the detection of all types of transitions is very important.

The twin-comparison algorithm is very sensitive to object motion, the appearance or disappearance of objects, camera panning and zooming etc. In the original paper [3] the problem is solved by passing detected transitions to the Motion Transition Removal test. In the proposed algorithm we introduce a combination of the twin-comparison algorithm and a feature presented in [6]- the absolute difference between two frames in their ratios of dominant (grass) colored pixels to total number of pixels denoted by G_d. Computation of G_d between i*th* and (i-k)*th* frames is given by (17)

$$G_d\left(i,k\right)=\left|G_i-G_{i-k}\right| \tag{17}$$

Using the grass colored pixel ratio we wanted to reduce the number of false positives in gradual transition detection. Our approach is based on the assumption that frames in two adjacent shots (before and after the gradual transition) often have a noticeable difference in their G_d.

The second pass in the twin-comparison algorithm detects gradual transition when the consecutive difference between frames falls below the lower threshold T_l and the cumulative difference (between starting frame F_s and F_e) exceeds the higher threshold T_h. We modify this step by using an additional criterion for the detection of the ending frame in a gradual transition:

$$G_d(F_s,F_e)>T_g \tag{18}$$

where $G_d(F_s,F_e)$ is the absolute difference in grass colored pixel ratio between starting frame F_s and ending frame F_e, and T_g is a threshold that is video dependent. Equation (18) sets an additional condition that starting frame F_s and ending frame F_e must satisfy: their difference in grass colored pixel ratio must be higher than threshold T_g. With this approach we wanted to reduce the sensitivity of the twin-comparison algorithm to object and camera motion. For example, a typical situation in soccer video is player tracking using fast camera panning, especially in close-up shots (above-waist view of person). In this case histogram differences between frames are higher than low threshold T_l and the twin-comparison method will often generate a false positive. The grass colored pixel ratio doesn't change so significantly because the shot usually displays one view (e.g. global view of the field). Therefore, checking the difference in G_d will eliminate some false alarms.

5. Results

We have tested the proposed algorithm on a data set of approximately 30 minutes of soccer video. The test set is composed of 3 MPEG-2 clips (resolution 352*288 at 25 fps) containing 85 cuts and 38 gradual transitions detected by a human observer. The algorithm is implemented using the Intel Open Source Computer Vision Library [8].

A highly useful way to measure and compare the effectiveness of different algorithms is to compute their recall (R) and precision (P):

$$R=\frac{Correct}{Correct+Missed}\times 100$$

$$P=\frac{Correct}{Correct+FalsePositives}\times 100 \tag{19}$$

In table I recall and precision rates are given for the standard twin-comparison algorithm. Table II shows results for our combination of twin-comparison algorithm and dominant colored pixel ratio. The performance of the algorithms for cut-type boundaries and gradual transitions is tabulated separately.

Both algorithms use the same method for cut detection (the first pass of the twin-comparison algorithm without G_d) and they have relatively high recall and precision rates for cut-type boundaries (table I and table II).The twin-comparison algorithm has a low precision rate (48%) for gradual transitions due to its well known sensitivity to camera and object movements. Our algorithm generates a higher precision rate (62%) because the criterion described with equation (18) eliminates some false positives.

However, for gradual transitions our algorithm has lower overall recall (42%) than the twin-comparison algorithm (63%) because criterion (18) in addition to false positives also eliminates some true gradual transitions. This problem occurs when we have a gradual transition between two shots with similar grass ratio. Such transitions don't pass the criterion for $G_d(F_s,F_e)$ and they are discarded as false positives. By lowering threshold T_g the recall rate is improved, but simultaneously the number of false alarms grows and precision is lowered. Thus, it is important to find the appropriate threshold.

Sequence	1		2		3		All	
Length	4:40		6:40		5:00		16:20	
Type	C	G.T.	C	G.T.	C	G.T.	C	G.T.
Correct	37	8	26	9	12	7	75	24
False	2	3	4	15	7	8	13	26
Miss	1	4	7	3	2	7	10	14
Recall	97	67	79	75	86	58	88	63
Precision	95	73	87	38	63	47	85	48

Table 1. Shot boundary detection results for twin-comparison algorithm [3] (C=cut, G.T.= gradual transitions)

Sequence	1		2		3		All	
Length	4:40		6:40		5:00		16:20	
Type	C	G.T.	C	G.T.	C	G.T.	C	G.T.
Correct	37	6	29	5	12	5	78	16
False	2	2	4	4	8	4	14	10
Miss	1	6	4	7	2	9	7	22
Recall	97	50	88	42	86	42	85	42
Precision	95	75	88	56	60	56	92	62

Table 2. Shot boundary detection results for proposed algorithm (C=cut, G.T.= gradual transitions)

6. Conclusion

In this paper a combination of the twin-comparison algorithm and dominant colored pixel ratio was introduced. For gradual transition detection our algorithm generates fewer false alarms than the standard twin-comparison algorithm and hence it has higher precision. During camera and object movements the difference in grass colored pixel ratio doesn't exceed threshold T_g and this results with less false detections. However, the standard twin-comparison algorithm has higher recall because our algorithm eliminates some true gradual transitions.

Future work will include improvement of the recall rate for gradual transitions by detecting camera operations that generate false alarms. For this purpose motion vectors from the MPEG bitstream can be used. Through motion vector analysis camera panning and zooming can be easily detected.

Acknowledgments

This work was supported by the Ministry of Science and Technology of the Republic Croatia under projects: ICT systems and services based on integration of information (023-0231924-1661) and Computer Vision in Identification of Sport Activities Kinematics (177-0232006-1662)

References

[1] I.Koprinska and S.Carrato, Temporal Video Segmentation: A Survey, *Signal Processing Image Communication, Elsevier Science*, 2001.

[2] Kikukawa, S. Kawafuchi, Development of an automatic summary editing system for theaudio-visual resources, *Transactions on Electronics and Information J75-A* (1992) 204-212.

[3] H.J. Zhang, A. Kankanhalli, S.W. Smoliar, Automatic partitioning of full-motion video, *Multimedia Systems* 1(1) (1993) 10-28

[4] B. Yeo, B. Liu, Rapid scene analysis on compressed video, *IEEE Transactions on Circuits & Systems for Video Technology* 5(6) (1995) 533-544.

[5] K. Shen, E. Delp, A fast algorithm for video parsing using MPEG compressed sequences, *Proc. Intern. Conf. Image Processing (ICIP'96),* Lausanne, 1996.

[6] Ekin, A.; Tekalp, A.M.; Mehrotra, R., Automatic soccer video analysis and summarization, *IEEE Transactions on Image Processing* ,Volume 12, Issue 7, July 2003 Page(s):796 – 807

[7] J.S. Boreczky, L.A. Rowe, Comparison of video shot boundary detection techniques, *Proc. IS&T/SPIE Intern. Symposium Electronic Imaging*, San Jose, 1996.

[8] Open Source Computer Vision Library, www.intel.com/technology/computing/opencv

[9] V. Kobla, D. De Menthon, and D. Doermann, Special effect edit detection using VideoTrails: a comparison with existing techniques, *Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases* VII, 1999

[10] Jesús Bescós, Guillermo Cisneros, José M. Martínez, José M. Menéndez, Julián Cabrera, A unified model for techniques on video-shot transition detection, *IEEE Transactions on Multimedia*, Volume 7, Issue 2, April 2005 Page(s):293-307