A new method for soccer shot detection with multi-resolution DCT

Youness TABII[1] Rachid OULAD HAJ THAMI[1]

[1] Laboratoire SI2M, Equipe WiM

ENSIAS B.P 713
Université Mohamed V, Souissi, RABAT - Maroc.

youness.tabii@gmail.com, oulad@ensias.ma

Abstract

Shot is often used as a basic unit for video analysis and indexing. A shot may be defined as a sequence of frames captured by "a single camera in a single continuous action in time and space". The extraction of this unit (Shot) still present problems for sports video. In this paper, we propose a novel method for soccer shot detection using discrete cosine transform multi-resolution (DCT-MR). First, we detect the dominant color component by supposing that playfield pixels are green (dominant color). Second, we compute the discreet cosine transform with different resolution (from the lowest resolution to the highest). Finally, we use a prefixed threshold to distinguish different transitions among shots.

keywords

video soccer, shot detection, cut detection, binary image.

1 Introduction

Since football is the most popular game in the world, the analysis of sports videos, and football videos in particular, has become an important research field that has attracted many researchers. Video documents presents audio/visual information which makes it possible to analyze well this type of documents and extract the semantics of shots one from videos using several algorithms.

In order to achieve a reliable video description, the primary requirement is to structure the video into elementary shots. This structuration consists of detecting transition effects between homogeneous segments (shots). This video partitioning step enables us to provide content-based browsing of the video. This stage should facilitate higher level tasks such as video editing or retrieval.

For ease of reference, we have to provide brief definition of the different kinds of shots boundaries. A cut is an abrupt transition between two shots that occurs between two adjacent frames. A fade is a gradual change in brightness, either starting or ending with a black frame. A dissolve is similar to a fade except that it occurs between two shots. The images of the first shot get dimmer and those of the second shot get brighter until the second shot replaces the first one. Other types of shot transitions include wipes and computer generated effects such as morphing.

In this sub-section, we have present a number of related works which dealt with detection of boundary shots. In [1], Yeo and Liu proposed a local threshold technique within a sliding window of frame dissimilarity values by considering the largest and the second largest peak. There are many other approaches that use a sliding window technique [2][3]. In [2], Bescos analyzes several frame disparity functions, i.e. functions which measure frame dissimilarities. Deterministic (e.g. summation of absolute differences), statistic parametric (e.g. likelihood ratio test) as well as statistic non-parametric disparity functions are considered. The authors choose two metrics which undertake the best divergence between the "cuts" and "noncuts" classes and computes a third feature which uses a small sliding window of size 1. Furthermore, a simple supervised parallelepipedic classifier is applied. The following results are reported for a subset of the MPEG-7 test set (2074 cuts) : 99% recall and 95% precision. Chua et al. [3] propose a unified approach to detect cuts and gradual transitions by using a temporal multi-resolution approach. This method affected by applying a wavelet transform to frame dissimilarity measures. They use histogram differences as well as a coarse representation of MPEG motion vectors. First, they detect candidates from the set of local maxima and then apply an adaptive thresholding technique. Finally, they use support vector machines via active learning to find an optimal hyperplane to separate cuts and non-cuts.

Other methods are based on frame comparison (dissimilarity measure) such as pixel-by-pixel frame comparison [4], which gives good results but induces a very high complexity and it is not robust to noise and camera motion. As well as frame content representation by histograms and vector distances measures produce a good frame dissimilarity measure [5], since histograms lack of spatial information. They needs to be compensate with local histograms [6] or edge detection [7]. Pairwise comparison checks each pixel in one frame with the corresponding pixel in the next frame [8]. The Likelihood ratio approach compares blocks of pixel regions [8]. The Colour histogram method compares the intensity or colour histograms between adjacent frames [9]. The methods above work on uncompressed video, but some other approaches work on the compressed data itself. These methods utilize the compression features of MPEG for shot boundary detection. In [10] the authors use colour blocks in an MPEG stream to find shots. DCT-based shot boundary detection uses differences in motion encoded in an MPEG stream to find shots [11].

In [12] the authors present an algorithm to detect shot changes using the discreet cosine transforme(DCT). They calculate the DCT of the luminance matrix by block of 8x8, then the two distances between neighboring pixels(vertical and horizontal distance). The only threshold for the shot changes detection is that the average of vertical and horizontal distances is superior to 1/2.

In our case of football matches, the luminance is not a factor that we can use to detect shots. Especially in matches in which the playfield, climate and lighting conditions change from one match to another. For the motioned reasons we transformed into binary frames to limit the effects which can influence the results. After this stage, we calculate Discrete Cosine Transform DCT for every frames binary by block N x N (where N is the resolution) and we obtain the image of frames in the frequency domain. Then we compute horizontal and vertical distance between neighboring pixels at the edge of blocks.

Our algorithm is very fast in the case of abrupt shot detection because the calculations are only made at the first level of resolution. In case of failure, we move towards the superior level.

2 Playfield segmentation

In this section, we present the statistical computation of the dominant colour and binarization (playfield segmentation).

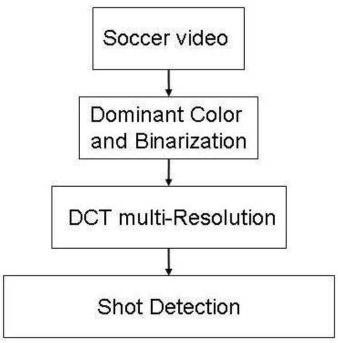


Figure 1 – *stages for shots detection.*

Figure 1 shows the different stages of our procedure for shot detection. The algorithm is composed of four steps : 1) Firstly, we select the video database for experiments our algorithm. 2) Secondly, we compute the dominant colour. The latter allows us to characterize the shots better. 3) Thirdly, we compute the discreet cosine transform for binarized frames using different resolutions. 4) Fourthly, once DCT is computed, we use it to detect the shots transition.

2.1 Dominant color extraction

The playfield usually has a distinct tone of green that may vary from stadium to stadium. But in the same stadium, this green colour may also change due to weather and lighting conditions (see Figure 2). Therefore, we do not assume any specific value for the dominant colour of the field.

(a) (b)

Figure 2 – *weather and lighting conditions*

We compute the statistics of the dominant field colour in the HSV space by taking the mean value of each colour component around its respective histogram peaks, i_{peak}. An interval $[i_{min}, i_{max}]$ is defined around each i_{peak}. The same method is adopted in [13] :

$$\sum_{i=i_{min}}^{i_{peak}} H[i] <= 2H[i_{peak}] \;\; and \sum_{i=i_{min}-1}^{i_{peak}} H[i] > 2H[i_{peak}] \tag{1}$$

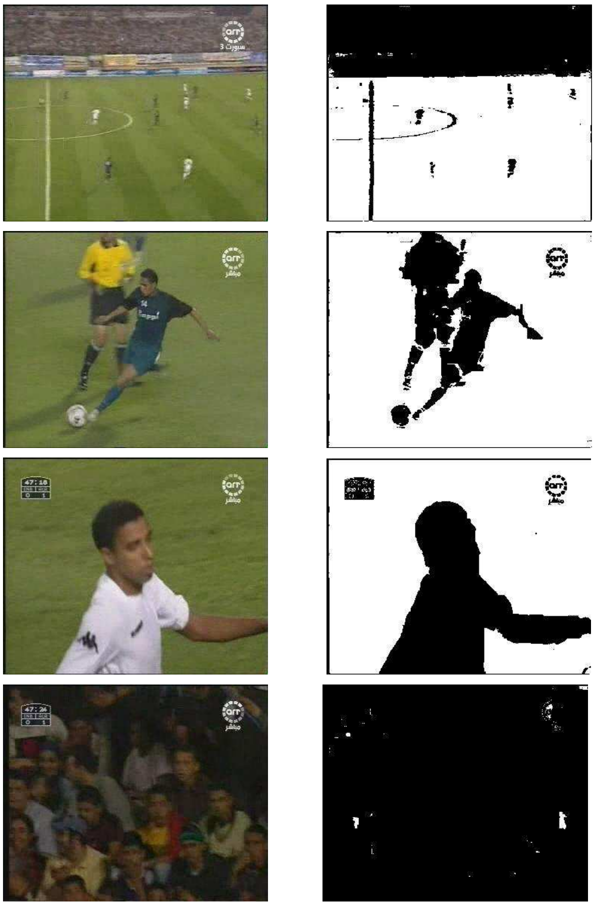
$$\sum_{i=i_{peak}}^{i_{max}} H[i] <= 2H[i_{peak}] \;\; and \sum_{i=i_{peak}}^{i_{max}+1} H[i] > 2H[i_{peak}] \tag{2}$$

$$colomean = \frac{\sum_{i=i_{min}}^{i_{max}} H[i] * i}{\sum_{i=i_{min}}^{i_{max}} H[i]} \tag{3}$$

Using the following quantization factor : 64 hue, 64 saturation, 128 intensity, H is the histogram for each colour component (H,S,V). Finally, the colour mean is then converted into (R_{mean}, G_{mean},B_{mean}) space so as to determine the playfield surface :

$$G(x,y) = \begin{cases} 1 \quad \text{if} \begin{cases} I_G(x,y) > I_R(x,y) + K(G_{peak} - Rpeak) \\ I_G(x,y) > I_B(x,y) + K(G_{peak} - Bpeak) \\ |I_R - Rpeak| < R_t \\ |I_G - Gpeak| < G_t \\ |I_B - Bpeak| < R_t \\ I_G > Gth \end{cases} \\ 0 \quad \text{otherwise} \end{cases} \tag{4}$$

G(x,y) is the binarized image frame in the field colour. In our system and after a number of tests, the new thresholds are : $R_t = 12$, $Gt = 18$, $B_t = 10$, $K = 0.9$ and $G_{th} = 85$. the Eq 1-3 are computed for every I/P frames.



(a) (b)

Figure 3 – *binarizarion : (a) original shot, (b) binarized shot*

3 Multi-Resolution DCT

In this section, we firstly define Discrete Cosine Transform. Then we present our DCT multi-resolution algorithm of for shot detection.

A discrete cosine transform(DCT) is a Fourier-related transform similar to the discrete Fourier transform(DFT), contrary to DFT, the projection core of DCT is a cosine and, thus, generates coefficients which are real. In DFT the core is a complex exponential which, therefore, generates coefficients which are complex. In domains related to the processing of image/video documents, DCT is more suitable to represent the set of points of a spatial domain in an equivalent representation in frequential domain with real numbers so as to facilitate the analysis and deduction.
The DCT operates on an A block of M*N image samples or residual values after prediction and creates B, witch is an M*N block of coefficients :

$$\begin{cases} B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn} \cos \frac{\Pi(2m+1)p}{2M} \cos \frac{\Pi(2n+1)q}{2N} \\ 0 <= q <= N-1 \\ 0 <= p <= M-1 \\ \alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 <= p <= M-1 \end{cases} \\ \alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 <= q <= N-1 \end{cases} \end{cases} \tag{5}$$

where M and N are the row and column size of A, respectively.

For each key frame, we compute the DCT with different resolutions, from the lowest resolution to the highest and for each resolution a we make a prefixed threshold to detect various transition between shots. Note that for each kind of transition has also a prefixed threshold. So for every type of transition has one threshold according to the resolution.

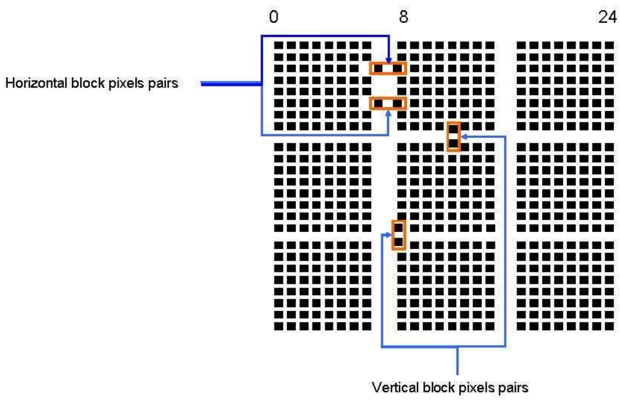


Figure 4 – *presentation of adjacent(Horizontal and Vertical)pixels* .

The vertical distance between adjacent pixels of key frames can be defined as :

$$distV = \sum_{i=1}^{\frac{w-R}{R}} \sum_{j=1}^{h} \frac{| \, pixel_{Rij} - pixel_{R(i+1)j} \, |}{h(w-R)/R} \qquad (6)$$

Similarly, the horizontal distance is :

$$distH = \sum_{i=1}^{w} \sum_{j=1}^{\frac{h-R}{R}} \frac{| \, pixel_{iRj} - pixel_{iR(j+1)} \, |}{w(h-R)/R} \qquad (7)$$

Thus, the mean distance for all adjacent pixels of video key frame with width *w* and height *h* is calculated as :

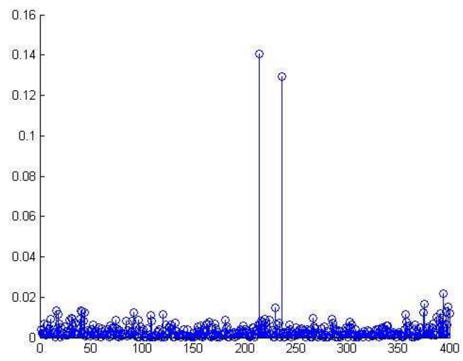
$$distHV = \frac{distH + distV}{2} \qquad (8)$$

where **R** is the resolution. For all key frames we compute distH, distV and distHV using Equations : 6, 7, 8

For abrupt transition, our algorithm computed the DCT with lowest resolution, which makes our algorithm very fast in detecting this kind of scene transition. For other types we need to do calculations for the next level resolution.

4 Experimental result

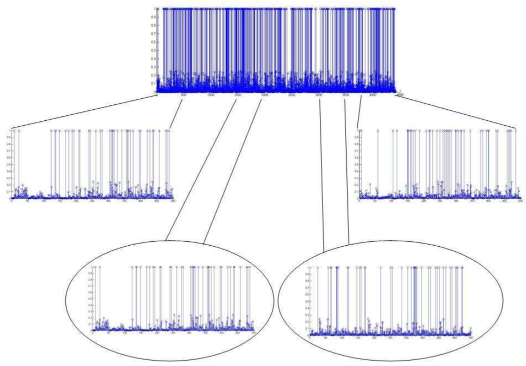
The video sequences used for the evaluation of the shot detection algorithm are MPEG compressed movies. This will allow us to test our algorithm on MPEG artificats due to the compression. The sequences also contain objects and camera motion. About 6 hours video sequences of various soccer matches in different champions leagues transcoded into MPEG 352x288, 1150kbps are used.



(a)

Figure 5 – *result for 400 I/P frames with height resolution R=5*

The threshold used in our case is $Thv = 0.12$ for different resolutions.
Tables 1 and 2 show the result we obtained for abrupt shot detection. The detection rate . This is may be due to the pre-fixed thresholds In other words, the features are less discriminant for this types of shots. However our algorithm works satisfactorily.



(a)

Figure 6 – *result for 5000 I frames with height resolution R=5(if $distHV > Thv$ so distHV=1)*

Tableau 1 – *Results of shot detection algorithm R=1,2*

Resolution	R=1		R=2	
	TSR1	FSR1	TSR2	FSR2
TTr : 566/5000 shots	212/566	87/566	147/354	72/354

Tableau 2 – *Results of shot detection algorithm R=3,4,5*

R=3		R=4		R5	
TSR3	FSR3	TSR4	FSR4	TSR5	FSR5
89/207	52/265	66/118	37/199	52/52	4/52

The TTr is the total of transitions per total of frames. the TSRi is True Shot in Resolution i, and FSRi is False Shot in Resolution i.

Conclusion

In this paper we presented a new method for the shot detection in video soccer on the basis of spatial analysis. The main contribution of the presented work is an algorithm for abrupt shots detection .

The advantage of our algorithm is clearly seen in its simplicity and effectiveness in providing better results for the detection of the majority of shots. Besides, the analysis of soccer video on the basis of playfield segmentation is very promising.

Références

[1] B. Yeo et B. Liu. Rapid scene analysis on compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(6) :533–544, 1995.

[2] J. Bescos. Real time shot change detection over online mpeg-2 video. *IEEE Transactions on Circuits and Systems for Video Technology*, 1(4) :475–484, April 2004.

[3] T.S Chua, H.M. Feng, et C. Anantharamu. An unified framework for shot boundary detection via active learning. *Proc. of ICASSP*, 2 :845–848, 2003.

[4] J.S. Boreczky et L.A. Rowe. A comparison of video shot boundary detection techniques. *Journal of Electronic Imaging*, 5(2), April 1996.

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

[6] A. Nagasaka et T. Tanaka. Automatic video indexing and full-video search for object appearances. *In IFIP Proc. of Visual Database Systems*, pages 113–127, 1992.

[7] R.Lienhart. Reliable transition detection in videos : A survey and practitioner's guide. *International Journal of Image and Graphics (IJIG)*, 1(3) :469–486, 2001.

[8] H. Zhang, S.W Smoliar, et H.J Wu. Content-based video browsing tools. *In Multimedia Computing and Networking*, pages 389–398, 1995.

[9] P. Bouthemy, C. Garcia, R. Ronfard, et G. Tziritas. Scene segmentation and image feature extraction for video indexing and retrieval. *In Visual Information and Information Systems*, pages 245–252, 1996.

[10] Y. Boon-Lock et B. Liu. A unified approach to temporal segmentation of motion jpeg and mpeg compressed video. *International Conference on Multimedia Computing and Systems, IEEE, Los Alamitos, Ca, USA*, pages 81–83, 1995.

[11] F. Arman, A. Hsu, et M. Chiu. Feature management for large video databases. *International Conference on Multimedia Computing and Systems, IEEE, Los Alamitos, Ca, USA*, pages 2–12, 1993.

[12] H. Koumaras, G. Gardikis, G. Xilouris, E. Pallis, et A. Kourtis. Shot boundary detection without thershold parameters. *Journal of Electronic Imaging*, 15(2), April-Jun 2006.

[13] W. Kongwah, Y. Xin, Y. Xinguo, et X. Changsheng. Real-time goal-mouth detection in mpeg soccer video. *ACM, Berkley, California, USA*, 2003.