MULTI TEMPORAL DISTANCE IMAGES FOR SHOT DETECTION IN SOCCER GAMES

Martin Hoernig, Michael Herrmann, Bernd Radig

Image Understanding and Knowledge-Based Systems
Technische Universität München, Boltzmannstrasse 3, 85748 München, Germany
{hoernig, herrmmic, radig}@in.tum.de

ABSTRACT

We present a new approach for video shot detection and introduce multi temporal distance images (MTDIs), formed by chi-square based similarity measures that are calculated pairwise within a floating window of video frames. By using MTDI-based boundary detectors, various cuts and transitions in various shapes (dissolves, overlayed effects, fades, and others) can be determined. The algorithm has been developed within the special context of soccer game TV broadcasts, where a particular interest in long view shots is intrinsic. With a correct shot detection rate in camera 1 shots of 98.2% within our representative test data set, our system outperforms competing state-of-the-art systems.

***Index Terms*—** soccer video analysis, video indexing, multi temporal distance image (MTDI), video segmentation, video shot boundary detection

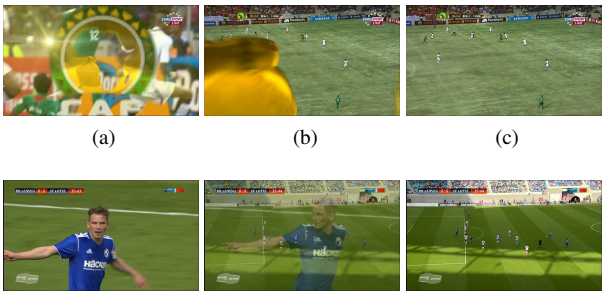

Fig. 1. Two exemplary transitions. (a)-(c) effect overlay. (d)-(f) dissolve transition. Frames from camera 1 (long view) are shown in images (c) and (f).

1. INTRODUCTION

A video is usually a combination of shots from different cameras. A shot itself is defined as a continuous and uninterrupted image sequence. Shot boundaries are formed by cuts or transitions (in the shape of dissolves, fades, wipes, effects, etc.). Fig. 1 shows examples. Camera flashes may also occur during shots and must be distinguished from shot boundaries. The challenge of boundary detection, and thus shot recognition, is the subject of this paper.

Good overviews of shot boundary detection are found in [1] and the earlier [2]. Both explain that transition detection is a more complex challenge than cut detection due to its higher temporal dependence and possible motion interference. Histogram-based approaches are discussed in [3, 4], while [5] incorporates statistics of gray value and edge-based differences. Other authors make use of entropy and SURF descriptors, e.g. [6], or spatio-temporal images, e.g. [7].

Systems developed for soccer applications include [8, 9]. Such approaches make use of grass detectors for general shot change detection and are significantly more context dependent than others.

By contrast, our paper tackles the robust detection of a relevant subset of shot boundaries, namely those of the main camera, *camera 1*. This camera records long views of the game (cf. Fig. 1 (c) and (f)) and thus provides images that are useful for a variety of analyses, e.g. [10]. Unlike news broadcasts or movies, camera 1 shots contain images with similar content (i.c. histograms) even during fast camera motions. Because of this as well as special effects at shot boundaries (see Fig. 1), finding these shots is a different problem than it is for general videos.

Our approach is inspired by [3] and [7]. We create feature matrices from histograms, as [3] suggests, but we compose them in a different manner, and in contrast to [7], no direct local dependency is incorporated or necessary. The result is a robust and fast system with low sensitivity to fast camera motions that models cuts and transitions (as well as effects) into shot boundaries.

2. SYSTEM OVERVIEW

Our approach consists of a new frame-wise shot change detection. After a color conversion, image-describing features allow us to calculate distances between time-adjacent frames. Based on these distances, frames are classified into in-shot, cut-in, cut-out and transition. Then, a shot merger transforms the video timeline into shot intervals based on the frame classes. In addition to its quantitative assumptions (see Table 1), our model also incorporates three qualitative assumptions: the main action takes place in the center of the screen, the director will not set a cut or a fade effect between two non-differentiable frames, and the soccer video is full screen without picture-in-picture methods.

Variable	Default	Meaning
T_s	$2s$	Minimum shot duration
T_t	$1.5s$	Maximum transition duration

Table 1. Quantitative model assumptions

3. SHOT DETECTION

The images of the input video (with constant size within the video) are first converted into an appropriate colorspace. We use CIELAB color space because the distances approximate human perception and the resulting histogram matrices (M^i, introduced in Section 3.1) are populated primarily by zeros in our scenarios. The usage of sparse matrices improves the computational cost.

3.1. Feature generation

Every image is divided into a constant decomposition of subimages, convex combined with weights $w_\bullet$. To simplify calculation, we relax the problem to rectangular subimages of equal size. The area not covered due to residuals can be cropped. Usually the image size is much higher than the number of rectangles, so the residuals carry no weight and can be left out. As an example, our default decomposition is shown in Fig. 2. The weights are set *a priori* according to an expected amount of regional importance. This models the first qualitative assumption.

$1\,(\frac{1}{20})$	$2\,(\frac{1}{20})$	$3\,(\frac{1}{20})$	$4\,(\frac{1}{20})$
$5\,(\frac{1}{20})$	$6\,(\frac{2}{20})$	$7\,(\frac{2}{20})$	$8\,(\frac{1}{20})$
$9\,(\frac{1}{20})$	$10\,(\frac{2}{20})$	$11\,(\frac{2}{20})$	$12\,(\frac{1}{20})$
$13\,(\frac{1}{20})$	$14\,(\frac{1}{20})$	$15\,(\frac{1}{20})$	$16\,(\frac{1}{20})$

Fig. 2. Decomposition of an image into rectangles with column indices of the feature matrix M^i and default weights $w_\bullet$ in parentheses.

Let us choose an arbitrary video image with frame index i (image #i). For each cell (or rectangle) a normalized 3D histogram on the underlying image data is calculated. By reducing the image to three bits for each channel, $(2^3)^3 = 512$ bins are used. Features can now be inserted into 512-tuples containing the histogram information. One 512-tuple for every cell is created, in the case of Fig. 2, a total of 16 tuples with 512 entries are created. They are inserted column-wise into the matrix M^i, the feature matrix for image #i. The histograms are normalized separately such that the sum of the $(512 * 16)$ matrix entries is equal to the number of columns (16).

3.2. Distance calculation

We use the χ^2-measure (see [11]) to calculate distances between images. This has two advantages: we can directly obtain a symmetric distance and the value is restricted to the interval $[0, 1]$. Of course, the measure is not a metric because the coincidence and subadditivity (triangle inequality) axioms are unfulfilled. Nevertheless, we call it a distance. Adapting for weighted feature sets (in the feature matrices M^i and M^j of images #i and #j) with weights $w_\bullet$ results in:

$$\chi^2(M^i, M^j; w_\bullet) := \frac{1}{2} \sum_q w_q \sum_r \frac{\left(M^i_{r,q} - M^j_{r,q}\right)^2}{M^i_{r,q} + M^j_{r,q}}. \quad (1)$$

Addends with value $\frac{0}{0}$ are omitted. It is easy to show that $\chi^2(M^i, M^j; w_\bullet)$ is also symmetric and restricted to $[0, 1]$ because the histograms are normalized and the weights $w_\bullet$ sum to one. This distance enables a simple cut detection:

$$\chi^2(M^i, M^{i+1}; w_\bullet) \begin{cases} \geq \textit{threshold}, & \text{cut or flash at } i, \\ < \textit{threshold}, & i \text{ is within shot.} \end{cases} \quad (2)$$

The Recognition And Vision Library (RAVL) [12] uses a related system for shot boundary detection. Although cut detection is viable with such an approach, transitions need a more time-dependent approach.

3.3. Multi temporal distance images

We now consider distances between a broader range of images. Let $D := (d_{i,j})$ with $d_{i,j} := \chi^2(M^i, M^j; w_\bullet)$ be the matrix of distances. This matrix is real and symmetric, its elements are nonnegative, and all diagonal elements are zero.

This matrix becomes very large (the square of the number of video frames), however, data outside a diagonal neighborhood are not of interest. To reduce memory and calculation cost, an online approach with smaller matrices is introduced. The necessary neighborhood size is $z := \lfloor \frac{1}{2} T_t * \textit{fps} \rfloor$, where *fps* is the number of frames per second.

$$\hat{D}(i) := \begin{pmatrix} d_{i-z,i-2z} & \cdots & d_{i-z,i-z} & \cdots & d_{i-z,i} \\ \vdots & & \vdots & & \vdots \\ d_{i,i-z} & \cdots & d_{i,i} & \cdots & d_{i,i+z} \\ \vdots & & \vdots & & \vdots \\ d_{i+z,i} & \cdots & d_{i+z,i+z} & \cdots & d_{i+z,i+2z} \end{pmatrix} \quad (3)$$

We call these matrices *multi temporal distance images (MTDIs)*, since every entry can be interpreted as an image gray value. The center, with coordinates $(z + 1, z + 1)$, is its *reference point* (with value 0). To obtain $\hat{D}(i + 1)$, z distance calculations (for $d_{i+z,i+z+1}, \ldots, d_{i+z,i+2z}$) must be executed if $\hat{D}(i)$ is known.

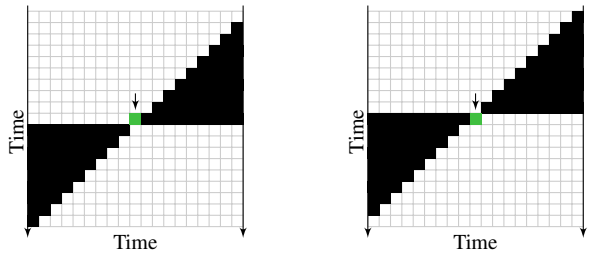

Fig. 3. MTDIs (shown on a grid) for cut-in (left) and cut-out (right) for two shots with maximal inter shot distance and minimal intra shot distance. The reference point is indicated by an arrow.

3.4. Cut detection

Cuts are divided into two different events: the cut-in is the last frame of a shot before a cut occurs and the cut-out is the first frame after a cut. As shown in Fig. 3, we use stair shapes to build a detector. This allows for a more robust classification than could be achieved using a frame-to-frame distance. The upper stair shape for cut-in detection includes every MTDI entry $\hat{d}_{r,q}$ with $r \leq z+1$ and $q > 2z+1-r$ with reference point at coordinates $(z+1, z+1)$. The lower stairs as well as the cut-out detection are redundant because the values within the lower stairs all together appear in the upper stairs. A frame is classified positive if all entries from the upper stairs (Fig. 3, left) are element-wise above a selected threshold value, denoted B_c.

3.5. Transition detection

A more complicated scenario is the detection of transitions: the shot change can be embedded into an effect and the transition length is unknown. An example of an ideal transition (transition time is equal to T_t, inter shot distances are maximal, and there are no intra shot distances) and a real world example (a shorter transition length, with interference) is shown in Fig. 4. To overcome the task of detecting transition centers, the results of four individual MTDI classifiers are merged.

Shape model. Transitions form noticeable shapes within MTDIs. If the duration is reduced to one intermediate frame between cut-in and cut-out, the resulting structure (in the ideal case) is similar to that of the cut (see Fig. 3). The intermediate frame appears as a constant horizontal line between the two stair shapes. The resulting shape is used as a template MTDI $T = (t_{i,j})$ with respect to equal neighborhood size z between template and video image. The construction of T is analogous to the stair construction within Section 3.4 except that now every covered entry is set to one and uncovered entries are set to zero. Template compliance can be measured using a threshold value B_s (like cut threshold B_c) by

$$p_1(\hat{D}(i)) := \frac{\#\{(r,q) \mid \hat{d}_{r,q} * t_{r,q} \geq B_s\}}{\#\{(r,q) \mid t_{r,q} = 1\}}. \qquad (4)$$

Sum model. Transitions longer than one frame lead to high values of p_1 at frames within a neighborhood of the actual transition center. This issue is addressed by two models,

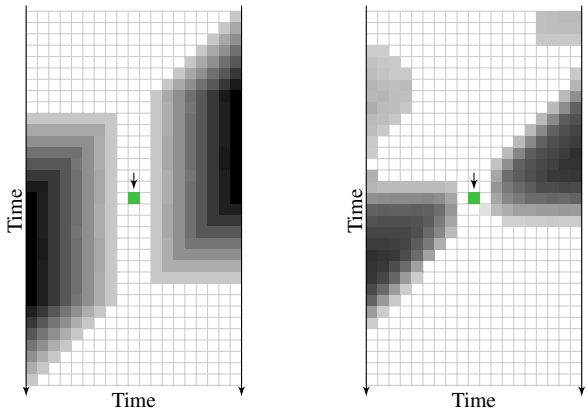

Fig. 4. MTDIs for transitions within an ideal scenario (left) and the real world scenario from Fig. 1 (d)-(f) (right). The gray values on the left are scaled by a factor of two for better visibility. The MTDIs are vertically extended to offer better insight.

the sum model and the weight model. The former uses the fact that the distance sum in every MTDI line should monotonically decrease in both temporal directions from the transition center.

$$p_2(\hat{D}(i)) := \frac{1}{2z} * \left(\#\left\{ r \;\middle|\; 1 \leq r \leq z, \textstyle\sum_q \hat{d}_{r,q} \leq \sum_q \hat{d}_{r+1,q} \right\} + \#\left\{ r \;\middle|\; z+1 \leq r \leq 2z, \textstyle\sum_q \hat{d}_{r,q} \geq \sum_q \hat{d}_{r+1,q} \right\} \right) \qquad (5)$$

Weight model. The sum model penalizes jumps in the sequence of line sums only once. The weight model uses the fact that the transition center frame is the MTDI row with highest line sum within its neighborhood. All line sums $\sum_q \hat{d}_{r,q}$ with $r \neq z+1$ are sorted by increasing value. The index of the first sum that is bigger than the line sum $\sum_q \hat{d}_{z+1,q}$ is denoted as j. If no list item fulfills this requirement, j is set to $2z+1$.

$$p_3(\hat{D}(i)) := \left(\frac{j-1}{2z} \right)^4 \qquad (6)$$

The exponent of four is used to promote the values near the end of the list. Frames with high p_3 values show the most dynamic actions in the underlying video sequences.

Differential model. This model processes the MTDI horizontal line at the reference point only. The line is numerically differentiated using the central difference quotient, resulting in a list with $2z-1$ entries. All elements are replaced by their absolute values. The last $z-1$ entries are considered to be function samples, measured equidistant from 0 to 1. A swing

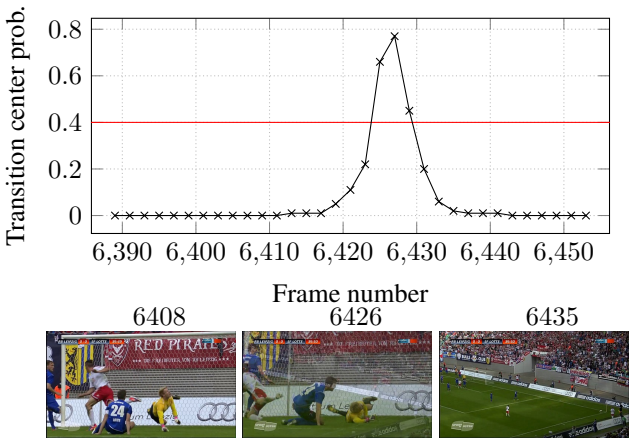

Fig. 5. Transition center probabilities within a test video (25fps). A transition (around frame #6427) causing a signal peak.

function $f(x) = \frac{a}{2}\left[\tanh(\pi(1-2bx))+1\right]$ with $0 < a \leq 1$ and $b \geq 1$ is now fitted to the sampled values, resulting in optimal parameters a_r and b_r. The same procedure is repeated for the left side, the first $z-1$ entries, and the swing function $f(-x+1)$, giving parameters a_l and b_l.

So far we have proposed a structure that is able to model the past and future distances from the reference point with four variables. The symmetry of frame distances at transition centers is the main feature we take advantage of. We now swap a_r and b_r with a_l and b_l and calculate the differences $\{v_1, \ldots, v_{2(z-1)}\}$ from each sample point to $f(x)$ and, respectively, $f(-x+1)$. The fourth detector

$$p_4(\hat{D}(i)) := 1 - \frac{1}{2(z-1)} \sum_r |v_r|^{0.5} \qquad (7)$$

therefore models the shape of the transition center.

Composing. The presented transition detectors are finally combined together to form a robust detection system.

$$P(\textit{transition center in frame no. } i) := \prod_{r=1}^{4} p_r(\hat{D}(i)) \qquad (8)$$

We call its output transition center probability. The threshold B_t now allows to decide if a frame can be classified as transition center. Fig. 5 shows an example.

3.6. Shot merger

Until this point, only frames have been classified. To retrieve shot intervals, all detected cuts and transition points (frames are associated with timeline points) as well as the start and end points of the video are considered to be shot boundaries. Because transitions are marked at their center points, half of the maximum transition duration T_t is once added and once subtracted from the center point timeline position and the range within is discarded. The remaining ranges and the resulting boundaries form the final shot intervals. Shots with a length shorter than T_s are not considered.

4. IMPLEMENTATION AND EVALUATION

Our shot detection system is based on only three thresholds B_c, B_s and B_t that have to be chosen. An example of the temporal behavior of the transition probability within a data set is shown in Fig. 5. Because the selected thresholds are not prone to interferences, we determined them empirically.

4.1. Evaluation data

We used the videos of five TV soccer matches for both training and testing our system. To test the ability of the system to adapt to different scenarios, we applied the model to games from different leagues, cups and continents. An overview of our test data is shown in Table 2.

As we are interested in finding long view shots from camera 1, we annotated every boundary from such a shot manually. If transitions were included, the boundaries were set at the final frame that showed image data from the corresponding shot only. Shots with a length shorter than three seconds were not considered.

4.2. Evalutation setup

Beside the MTDI system (with parameters $B_c = B_s = 0.2$, $B_t = 0.4$) we also evaluated two publications important once for shot detection in general and once specifically for shot detection in TV soccer videos:

The Fraunhofer Heinrich Hertz Institute's (HHI) model [5] is based on pixel, edge and histogram difference statistics. It consists of separate detectors for hard cuts and the transition types. The results achieved at the TRECVID conference [1] showed very good detection performance [5].

Secondly, we implemented the approach from Ekin et al. [8]. In contrast to the newer HHI concept, this system was built specifically for soccer shot detection. The thresholds in this system are trained during a separate learning stage. Screenshots from camera 1, closeups (100 from each class), and grass calibration images for every video were manually selected according to the paper guidelines.

Because Ekin's approach as well as the available version of the HHI system do not model transition lengths, we shortened every shot by one second at both boundaries. Without this step, every non cut shot change would result in a false detection if it concerned the boundary of a camera 1 shot. An annotated shot is now classified as detected if at least one shot change is detected within its first $1.5s$, at least one shot change is detected within its last $1.5s$, and no shot change is detected within $[start + 1.5s, end - 1.5s]$. This relaxation had no negative effect on the recognition rates of the systems. The shot merger of the MTDI model automatically shortens a shot if a transition is detected.

4.3. Results

The MTDI approach detects 98.2% of the camera 1 shots correctly. Errors are caused by missing shot boundaries between fades of similar images. No boundaries within camera 1 shots were detected (interrupted shot rate (ISR), measured against false detections: 0%).

Match (Video)	Length (mm:ss)	Shots	Cam1 shots
FC Bayern - Arsenal (UEFA Champions League, HD, night)	20:03	107	31
Burkina Faso - Ghana (Africa cup, many game interruptions)	55:56	620	123
M'Gladbach - Lazio (Europe League, inserts and ads included)	51:51	456	100
South Korea - Greece (World Cup 2006, midday sun)	3:00	43	12
RB Leipzig - SF Lotte (Forth tier of the German league system)	8:13	52	16
Σ	138:53	1278	282

Table 2. Information about the evaluated soccer games

Both Ekin's (ISR: 0.40%) and the HHI system (ISR: 0.30%) had difficulties with transition effects. In particular, the latter method suffers from false shot detections during rapid camera motions. The results are summarized in Table 3. Our approach shows the highest shot detection quote in every test video with a 100% score in three videos.

Match (Video)	MTDI	HHI	EKIN
FC Bayern - Arsenal	100.0%	87.1%	67.7%
Burkina Faso - Ghana	98.4%	87.8%	78.9%
M'Gladbach - Lazio	97.0%	87.0%	72.0%
South Korea - Greece	100.0%	100.0%	50.0%
RB Leipzig - SF Lotte	100.0%	93.8%	68.8%
Σ	98.2%	88.3%	73.4%

Table 3. Correctly identified camera 1 shots

4.4. Computational cost

If frame accuracy in shot boundary detection is not necessary, the system is able to skip frames without degrading performance. This results in smaller neighborhood sizes z and less computational cost. Using a fixed frame rate, the algorithms are frame-wise $\mathcal{O}(n)$ if n is the number of pixels per frame. This is due to the histogram creation. The distance and detector calculations are arithmetic operations of a fixed size and have a complexity of $\mathcal{O}(1)$. Implemented in C++ and executed on a Intel Core i7-2600k, we achieved a rate of 67.7 fps with HD data and 122.1 fps with SD data. Frame skipping at 15 fps, results in an improvement factor of 5.4 on HD50 videos and 9.8 on SD25 videos.

5. CONCLUSION

Our paper reveals that MTDI based shot boundary detectors show remarkable results, in both computational speed and correctness. Nearly every camera 1 shot boundary was detected correctly, independently from its type - transition (including effects) or cut. The established detector from the Fraunhofer HHI [5] as well as the soccer specific detector from Ekin et al. [8] were distanced regarding their detection performance.

REFERENCES

[1] Alan F Smeaton, Paul Over, and Aiden R Doherty, "Video shot boundary detection: Seven years of TRECVid activity," *Computer Vision and Image Understanding*, vol. 114, no. 4, pp. 411–418, 2010.

[2] John S Boreczky and Lawrence A Rowe, "Comparison of video shot boundary detection techniques," *Journal of Electronic Imaging*, vol. 5, no. 2, pp. 122–128, 1996.

[3] Ali Amiri and Mahmood Fathy, "Video Shot Boundary Detection Using QR-Decomposition and Gaussian Transition Detection," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 1, 2009.

[4] Swati D Bendale and Bijal J Talati, "Analysis of Popular Video Shot Boundary Detection Techniques in Uncompressed Domain," *International Journal of Computer Applications*, vol. 60, no. 3, pp. 30–33, 2012.

[5] Christian Petersohn, "Fraunhofer HHI at TRECVID 2004: Shot boundary detection system," in *TREC Video Retrieval Evaluation Online Proceedings, TRECVID*, 2004.

[6] J. Baber, Nitin Afzulpurkar, M.N. Dailey, and M. Bakhtyar, "Shot boundary detection from videos using entropy and local descriptor," in *Digital Signal Processing (DSP), 2011 17th International Conference on*, 2011, pp. 1–6.

[7] Chong-Wah Ngo, Ting-Chuen Pong, and Hong-Jiang Zhang, "Motion-Based Video Representation for Scene Change Detection," *International Journal of Computer Vision*, vol. 50, no. 2, pp. 127–142, 2002.

[8] Ahmet Ekin, A Murat Tekalp, and Rajiv Mehrotra, "Automatic soccer video analysis and summarization," *Image Processing, IEEE Transactions on*, vol. 12, no. 7, pp. 796–807, 2003.

[9] Shu-Ching Chen, Mei-Ling Shyu, Chengcui Zhang, Lin Luo, and Min Chen, "Detection of soccer goal shots using joint multimedia features and classification rules," *MDM/KDD*, vol. 3, 2003.

[10] M Alemán-Flores, L Alvarez, L Gomez, P Henriquez, and L Mazorra, "Camera calibration in sport event scenarios," *Pattern Recognition*, 2013.

[11] Ofir Pele and Michael Werman, "The Quadratic-Chi Histogram Distance Family," in *Computer Vision ECCV 2010*, vol. 6312 of *Lecture Notes in Computer Science*, pp. 749–762. Springer Berlin Heidelberg, 2010.

[12] C Galambos, "RAVL: Recognition And Vision Library.," http://ravl.sourceforge.net, 2000.