REAL-TIME CAMERA FIELD-VIEW TRACKING IN SOCCER VIDEO

Kongwah WAN, Joo-Hwee LIM, Changsheng XU, Xinguo YU
Laboratories for Information Technology
21 Heng Mui Keng Terrace, Singapore 119613
{kongwah, joohwee, xucs, xinguo}@lit.a-star.edu.sg

ABSTRACT

Soccer video content-based analysis remains a challenging problem due to the lack of structure in a soccer game. To automate game and tactic analysis, we need to detect and track important activities such as ball possession in a soccer video that is highly correlated to the camera's field-view. In this paper, we present a system that tracks the camera's field-view in a soccer video in real-time. It utilizes a host of content-based visual cues that are obtained by independent threads running in parallel. The result is visualized as an active rectangular bounding box that approximates the camera's field of view superimposed on a virtual soccer field. Experimental results show that the system can reliably track the camera field-view as the game progresses.

1. INTRODUCTION

The soccer game attracts a global viewer-ship and research effort has been focused on detecting key highlights to facilitate annotation and browsing. The bulk of the literature mainly lies in video structure analysis, general video segmentation and event/highlight detection. Shots are usually first classified using a variety of specific (e.g., lines, ellipse in [1,2,3]) and generic tools (e.g., dominant color ratios in [5]), followed by a model-based event detection algorithm [4,5,6]. To automate game and tactic analysis, we need to detect and track important activities such as ball possession in a soccer video which is correlated to the camera's field-view. In this paper, we focus on tracking the camera field-of-view with respect to the soccer field, in real-time. This is part of our overall project design to develop coaching tools to analyze the tactical aspects of soccer video. As shown in Figure 5, as the camera pans to follow the soccer ball during the game, our goal is to derive a good estimate of the physical field position of play.

In [1], Gong et al, has similarly defined nine positions of play covering the entire soccer field. Shots are first manually prepared and then classified as to which position of play it is in. The classification is based on low level visual features such as line markings, motion vectors, ball and players extracted from the frames. While, for obvious reason pertaining to the similar objective in our system, we reuse the same visual features, but we make the point that their proposed implementation is just too expensive for real-time usage. Therefore, faster algorithms are needed to achieve real time processing. Furthermore, to improve robustness, we also introduce additional visual features, and our processing is done at the frame-level.

We provide details of our system in Section 2. Section 3 tabulates our experimental results. We conclude by highlighting some issues and further work in Section 4.

2. SYSTEM ARCHITECTURE

The system flow is illustrated in Figure 1. The four dashed boxes indicate the four separate threads running, while the remaining modules are for synchronizing decisions and updating positions.

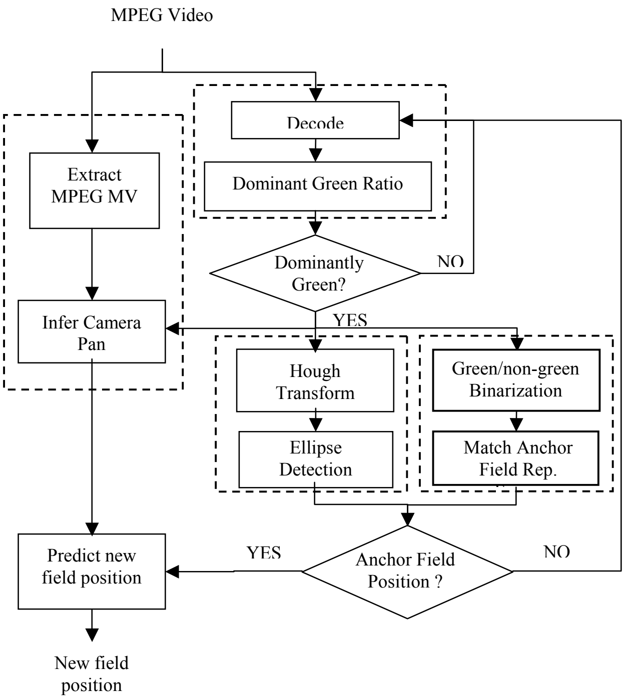

Figure 1: System Architecture

2.1. Dominant Green Ratio

The soccer video is not a homogenous contiguous single shot, but rather comprises of different shot segments from different cameras. Typically, the most important shot comes from the "grand-stand" camera, where a panoramic view of the field is captured, and where the camera also tracks and follows the ball. Clearly, our system relies on these shots to reliably track the field position of play in the game. According to our own observation on a variety of games collected from broadcast soccer video from the European League and also the recent World-Cup, these panoramic field-view shot comprises about 60% of the total video duration. If we were to include "following" shots wherein close-up of players dribbling the ball in play, we would have a sizable proportion (~70%) of video usable for our camera field-view tracking.

Similar to [1,2,3,5], we exploit the domain fact that the green field is the most dominant visual feature in a panoramic field-view shot. However, we have observed that due to the different playing ground (e.g., refurbished green turf in new stadium) and playing time (e.g., during a sunny day), the extent of the green hue is different in different soccer video. Hence, we take a random shot sample from the video to compute the dominant green hue deviation, and subsequently use that to distinguish the *green* pixels from *non-green* pixels in every video frame. This procedure of taking shot samples may correspond to an *initialization phase* in practical deployment.

2.2. Coarse Spatial Field Representation

Using the dominant green hue captured in Section 2.1, we sub-sample the original video frame into blocks of 32x32 and perform a simple 2-bin green/non-green quantization within each block. The resulting frame is a coarse spatial representation of the green field, from which we can quickly perform two tasks: (1) decide whether the current video frame is a panoramic view; and (2) estimate the vertical field position that the camera is focusing by counting the index of the first non-green line from the top. Figure 2 shows some examples.

2.3. Line Marks Detection

Once the current frame is deemed to be usable for field-view tracking, the system attempts to locate the line-marks often clearly visible on the soccer field. These line marks demarcate the field into natural playing sections such as the center field, left goal area, right goal area, etc.

We use a simple edge kernel to extract edge strength and feed them to a Hough Transform line accumulator [7]. We further constrain the angular transform to detect lines with orientations as observed on the left, center and right goal area. We point out that these may be obtainable during an initializing phase as mentioned in Section 2.1.

We also limit the edge extraction to the dominantly green regions within the video frame as detailed in Section 2.2. This has the advantage of not only reducing the Hough Transform computation, but also improving its performance by cutting down noises.

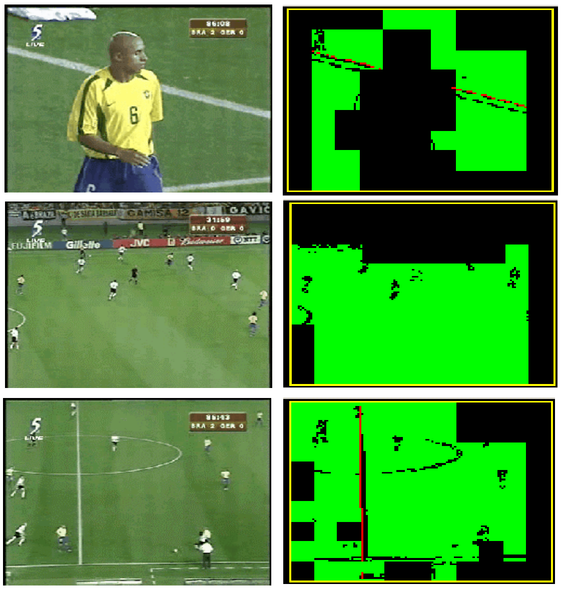

Figure 2: Coarse spatial representation of the field using dominant green hue binarization. The top row shows a typical representation for a close-up frame. The middle and bottom rows show the representations when the play is in the upper and lower boundary of the field respectively.

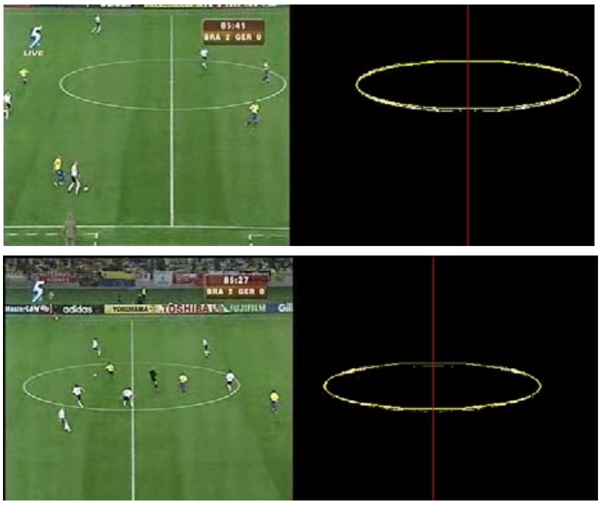

Figure 3: The location of a detected ellipse enables the system to tell whether the field of play is at the upper or lower center field.

Whenever a strong vertical line is detected, we further compute for the presence of an ellipse using the technique in [8]. As shown in Figure 3, not only does this strengthen the premise that the play is currently in mid-field, but it also provides greater granularity in telling whether it is the upper center field, or the lower center field. Since the ellipse-fitting algorithm is very sensitive to noise, we again extract edge points within the dominant green region. Furthermore, using the central vertical line as a guide, we filter out non-symmetrical edge points, and iteratively perform the ellipse fit. We evaluate the goodness of fit by its size and consistency over three consecutive frames.

2.4. Anchor Field Position Matching

It is clear that with the presence of simple visual features such as lines, the system can only process the field of play at a coarse granularity, e.g., left-field, center-field, right-field. To achieve finer granularity, there is a need for additional computation.

We observe that the sub-sampled two-bin color-quantized image (as detailed in Section 2.2) is visually dissimilar at the two ends of the field. Intuitively, this is caused by the row of billboards that also are served to demarcate the edge of the field. There is greater color variance in these pixel positions, and therefore will likely be quantized as non-green-dominant.

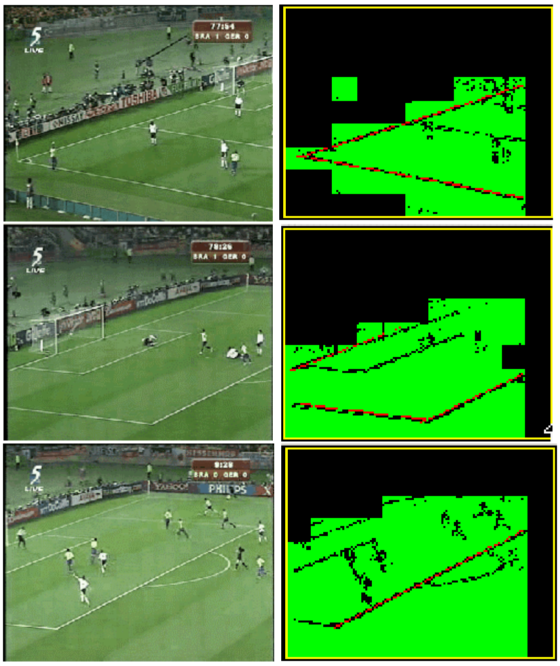

Figure 4: Sample Anchor field positions. In each row, the original video frame is on the left, with its corresponding color-binarized image on the right. The red lines denote the line marks detected by Hough Transform, and are also used during matching.

As shown in Figure 4, these features can be used to distinguish the field-of-play in the vertical y-axis, e.g., the lower left area, left-goal area and the upper-left area. We can think of these images as representative images at the various *anchor field positions (AFP)*. In our current implementation, we define six such AFPs to cover the two extreme ends of the field, three for the left goal area, and three for the right goal area. Figure 4 shows the representative images for the left goal area. Each representative image is selected during the system initialization phase. Though random video segments can be used, but care is taken to select the image frame where all key line marks are detected. This is because the Hough coefficients (ρ, θ) of these line marks are also used in the matching process.

Each representative image is sub-sampled and green/non-green-binarized in the exact manner is already described in Section 2.2, and stored in the memory as an AFP template. When a new image is presented for matching, it first undergoes the same normalization before a simple vector-similarity comparison with every AFP template. The matching score also takes into account the (ρ, θ) pair of every Hough line detected.

Furthermore, we do not encode AFP templates for the other remaining field positions such as center-field, in-between left etc. This is because the sub-sampled two-bin color-quantized image for all these positions are very similar. But we note that the line mark detection detailed Section 2.3 qualifies as some kind of AFP template for the center-field as well.

2.5. Inferring Camera Pan from Motion Vectors

A high matching score is imposed in AFP matching to ensure that the field position computed does not exhibit erratic behavior. One can think of a successful AFP match as a *reset* in the tracking process, or an I-frame in a predictive video sequence. In between resets, the system relies on the global MPEG motion vectors (MV) to estimate camera pan motion. These MVs are readily available in the compressed bit-stream. In our current implementation, the forward MVs are accumulated over three successive I/P-frames, and they are scaled to reflect the non-uniform camera pan with respect to the field view position. This is necessary because the camera pans a lot less vertically (than horizontally).

3. EXPERIMENT RESULTS

We tested our system on 2 MPEG-1 video digitized using the Hauppauge WinTV-USB card. The video encoding is done at a bitrate of 1.15Mbps, 352x288, 25fps.

We manually select shot sequences that show a variety of field-view movement that would intuitively correspond to different game tactics and team attack style. In order for the tracking to start reliably, we select shots that begin with a video frame near to an anchor field position (e.g., center-field where the vertical line and ellipse can be seen and detected, or at the two ends where the slant-oriented Hough line can also be seen and detected). The end of the shot should be a break (when the ball goes out of play). In between, the shots can be of a non-panoramic view like a "following" shot.

On each of these shot sequences, we label the "ground-truth" rectangular bounding box denoting the actual camera field-view. This box is set in order to be completely circumscribing the central ellipse. Figure 5 shows some labels. In total, we label 15 shots for two videos, totaling 721 ground truth labels. Each shot averages about 25 seconds. For each test video, a separate initialization needs to be done. A complete set of parameters saved include: the dominant green hue, the six AFP representative frames and their corresponding Hough coefficient pairs, the index of the first non-green line for a coarse spatial representation on a video frame depicting play at the upper boundary of the field.

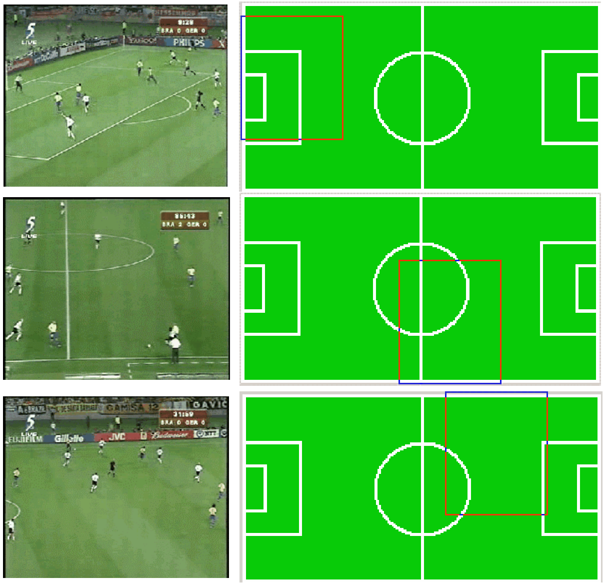

Figure 5: Ground truth labeling

We run our experiments on Pentium-4 1.7Ghz PC, with 128MB RAM, and the video source located on the harddisk. We used the publicly-available libmpeg2 decoder [9].

Table 1 shows the promising results obtained. The first column shows the result whereby the accuracy of each computed bounding box is compared against the ground truth at a 50% overlap ratio. The performance drops when a stricter overlap criterion is used.

Table 1: Test Results

	50% overlap	80% overlap
Video 1 (10 shots)	94% (452/481)	60% (288/481)
Video 2 (5 shots)	96% (230/240)	70% (168/240)

4. CONCLUSIONS AND FUTURE WORK

In this paper, we have explored developing a system that can automatically capture the camera's field-of-view with respect to the physical soccer field, in a soccer video. Our real-time requirements necessitates that all our visual feature extraction tools must be light-weight and run at frame-rate.

Though all the techniques used in this paper are simple at the core, the encouraging results demonstrate their feasibility in detecting and tracking the field position of play. With this, a query system can be immediately built to serve query such as: "Give me all the corner clips", "Give me all fast break clips". This is the immediate task for us. Our other work includes integrating ball and player tracking to improve tracking results.

5. REFERENCES

[1] Y.Gong, T.Lim, H.Chua, H.Zhang and M.Sakauchi, "Automatic parsing of TV soccer program," *Proc. 2nd Int. Conf on Multimedia Computing and Systems,* pp. 167-174, 1995.

[2] Y.Seo, S.Choi, H.Kim and K.Hong, "Where are the ball and players? Soccer game analysis with color-based tracking and image mosaick", *Proc. of Int'l Conf. Image Analysis and Processing (ICIAP'97)*, 1997.

[3] D.Yow, B.Yeo, M.Yeung and B.Liu, "Analysis and presentation of soccer highlights from digital video", *Proc. of 2nd Asian Conf. On Computer Vision (ACCV'95)*, pp.II499-503, 1995.

[4] V.Tovinkere and R.Qian, "Detecting Semantic Events in Soccer Games: Towards A Complete Solution", *Proc. ICME 2001*, pp 1040-1043, 2001.

[5] L.Xie, S.Chang, A.Divakaran, H.Sun, "Structure Analysis of Soccer Video with Hidden Markov Models", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2002.

[6] J.Assfalg, M.Bertini, A.Bimbo, W.Nunziati and P.Pala, "Soccer highlights detection and recognition using HMMs", *Proc ICME 2002*, pp , 2002.

[7] D.Ballard and C.Brown *Computer Vision*, Prentice-Hall, 1982, Chap. 4.

[8] Fitzgibbon, M.Pilu and R.Fisher ``*Direct least-square fitting of Ellipses*'', IEEE PAMI, pp 476-480, May 1999.

[9] http://libmpeg2.sourceforge.net/