INCREMENTAL RECTIFICATION OF SPORTS FIELDS IN VIDEO STREAMS WITH APPLICATION TO SOCCER

[1]*Jean-Bernard Hayet, Justus H. Piater and Jacques G. Verly*

[1]`Jean-Bernard.Hayet@ULg.ac.be`
Université de Liège, Institut Montefiore, B28, B-4000 Liège 1, Belgium

ABSTRACT

We describe a complete system for image rectification in sport video sequences. Relying both on geometrical properties of the field elements and on photogrammetric data, we compute an estimate of the field-to-image homography either incrementally – by inter-image data processing – or "from scratch" when direct image-model correspondences are possible. Examples are given in a soccer context.

1. INTRODUCTION

Soccer has become the most popular sport around the world. Every four years, hundreds of millions of fans enjoy their favourite sport thanks to television. The commercial aspect of sport TV broadcast is now enormous.
Broadcasters could take advantage of the latest techniques in computer vision, such as sequence analysis, 3D analysis, and uncalibrated vision. Indeed, extracting static (e.g., lengths) and dynamic (e.g., speed) metrics from single-camera images and using this information for automated sports analysis is essential for many applications. One example is the use of motion models that allow better tracking of players and of the ball by operating directly in the xy space of the field instead of in the image. Another example is the smart overlay of 3D objects, e.g., to overlay advertisements on the grass without obscuring players.
Sport broadcasters already use metrics extracted from single-camera images. However, current commercial systems for automatic tracking/analysis are very expensive. Indeed, most current systems rely on sensors embedded in cameras. They measure all camera parameters such as orientation in space and zoom to deduce the field-to-camera geometry. Some systems also use transmitters attached to players [1].
In this paper, we propose a fundamental building block that would be the heart of a future fully automated sports analysis system. This subsystem allows us to maintain a continuous estimate of the homographic mapping between images and a model of the field without using any kind of sensor on the cameras and the players [2].
This approach requires three critical elements: One for computing the initial homography or periodically recomputing it from scratch, one for quickly computing it from a reasonable guess and one for updating it from frame-to-frame correspondences. Periodic recalculation is required to initialize the system and to prevent drifts when incremental updating is done. The second procedure allows to attain real-time performance. Last, using frame-to-frame correspondences is necessary when image-to-model matches are not sufficient.
A similar idea has been proposed [3, 4], but is restricted to relatively narrow zones, e.g., just around the goal zones. This idea is also investigated in the case of hockey [5], but the field is much smaller so that many features remain visible for long periods of time, thereby making the task much easier.
This article is organized as follows. In Section 2, we give an overview of our system. Then, in Section 3, we briefly review the notion of homography, give notation and highlight important properties. Then, we give an overview of our system. Section 4 discusses the direct estimation of the homography based on feature correspondences. This is done in two different cases. Section 5 describes our line tracking algorithm. Section 6 shows experimental results we obtained from real soccer sequences. The last section draws conclusions and gives hints for future work.

2. OVERVIEW OF OUR SYSTEM

Figure 1 sums up the main elements of our system. Maintaining a continuous estimate of the homography throughout a whole sequence requires three main functionalities. First, the system must cope with model-to-image (M2I) correspondences. When a sufficient number of these correspondences are available, the homography can be estimated as described in the next section. This requires that features be extracted from the images and matched to corresponding features of the model. Feature extraction involves the development of a number of feature detectors, either generic or specific to the applications (such as lines and circles of a soccer field). Feature matching implies the definition of feature characterization and metrics. Note

Proceedings of the IEEE Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS'04), 2004.

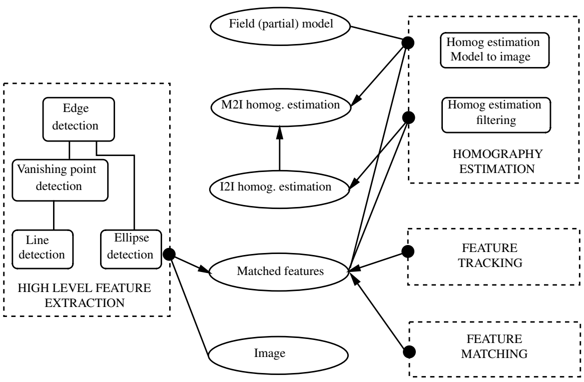
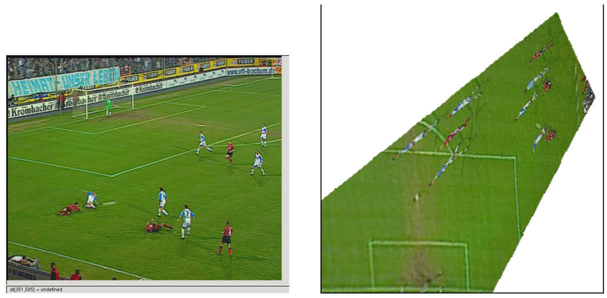


Figure 1: *Overview of our system.*



(a) (b)

Figure 2: *Illustration of the role of a homographic transformation. (a) Selected image in video stream. (b) Same image after rectification by homography.*

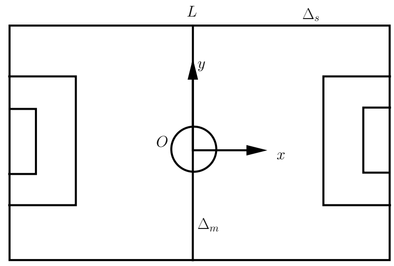


Figure 3: *The soccer field: Our conventions.*

that this operation is time-consuming.

Second, another component of the system is in charge of *tracking* features from image to image (I2I). Tracking by starting from the projected model allows to focus the search of features in a much smaller area, close to the estimated features position, so that speed is increased.

Last, the system must be able to cope with the situations where the correspondences in the tracking phase are not sufficient in number. To overcome this problem, the idea is to estimate the homography between two successive images. With such an approach, the model-to-image homography is incrementally updated.

Let us suppose that that $H_{i_0}^m$ is the model-to-image homography at a time i_0. For $i > i_0$, we propose to compute the inter-image homography $H_i^{i_0}$, so that:

$$\mathbf{H_i^m} = \mathbf{H_i^{i_0}} \mathbf{H_{i_0}^m}. \tag{1}$$

Estimating $H_i^{i_0}$ is done by using line segments and feature points. In this article, we focus only on the first two components: (1) estimating the homography from line correspondences and (2) tracking these correspondences along sequences.

3. COMPUTING THE HOMOGRAPHY FROM MODEL-TO-IMAGE CORRESPONDENCES

To explain more clearly the need for computing a field-to-image mapping, we can refer to Fig. 2. The left side shows an image taken from a soccer video sequence. The right side presents the result of the application of the computed field-to-image homography to the first image. The homographic transformation is such that Fig. 2(b) is independent of all camera parameters, as it compensates camera position, attitude, and internal parameters.

3.1. Notation

We represent all 2D points via homogenous coordinates. We use $(u, v, 1)^T$ for points p (lowercase) in each image and $(X, Y, 1)^T$ for points P (uppercase) in the 2D model, e.g., a 2D map of a soccer field. Matrices are written with bold characters to distinguish them from vectors and scalars.

The xy frame corresponding to the field model is shown in Fig. 3.

The model-to-image homography is represented by a 3×3 transformation matrix denoted by $\mathbf{H_i^m}$, where "i" stands for the sequence image we are dealing with and "m" for the model. It can be shown in that $\mathbf{H_i^m}$ is related to the rigid transformation $(\mathbf{R}, t)$ linking the camera frame and the xy frame by [2] :

$$\mathbf{H_i^m} \sim \mathbf{K}[r_1 r_2 t], \tag{2}$$

where the vectors r_1 and r_2 are the first two columns of the 3×3 matrix $\mathbf{R}$ and t is the column vector representing translation. Matrix $\mathbf{K}$ contains the camera internal parameters. We recall that "$\sim$" means "equals up to a scale factor".

The 3×3 matrix $\mathbf{H_i^m}$ will often be represented by an equivalent 9×1 vector denoted by h_i^m. It is obtained by stacking the successive rows of $\mathbf{H_i^m}$ on top of each other.

3.2. Determining $\mathbf{H_i^m}$ from point correspondences

The homography matrix $\mathbf{H_i^m}$ has 8 degrees of freedom ($9-1$ because of the scaling factor). Therefore, as each correspondence contributes 2 equations, a minimum of 4 point – or, equivalently, line – correspondences are necessary to estimate H_i^m. It is significant that the required correspondences can come from a wide variety of features extracted from the imagery, e.g., points, lines, circles.
Let us assume we have extracted K correspondences (P_k, p_k), $k = 1, \ldots, K$, where $P_k = (x_k, y_k, 1)^T$ and $p_k = (u_k, v_k, 1)$. This is mathematically expressed as:

$$\begin{aligned} \mathbf{H_i^m} P_1 &= p_1 \\ \mathbf{H_i^m} P_2 &= p_2 \\ &\vdots \\ \mathbf{H_i^m} P_K &= p_K. \end{aligned}$$

These equations can also be written in terms of the 9×1 vector h_i^m as:

$$\mathbf{A} h_i^m = 0, \tag{3}$$

where $\mathbf{A}$ is the $2K \times 9$ matrix including all the correspondences

$$\mathbf{A} = \begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -u_1 x_1 & -u_1 y_1 & -u_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -v_1 x_1 & -v_1 y_1 & -v_1 \\ \ldots \\ x_K & y_K & 1 & 0 & 0 & 0 & -u_K x_K & -u_K y_K & -u_K \\ 0 & 0 & 0 & x_K & y_K & K & -v_K x_K & -v_K y_K & -v_K \end{pmatrix}.$$

The overdetermined system of homogeneous linear Eq. 3 is easily solved in the least-squares sense, expressed as:

$$\min \|\mathbf{A} h_i^m\|^2, \tag{4}$$

by SVD decomposition of matrix $\mathbf{A}^T\mathbf{A}$. The least-squares solution h_i^m is then given by the singular vector of $\mathbf{A}$ corresponding to the smallest singular value. The homography is usually normalized by setting $(\mathbf{H_i^m})_{33} = 1$.

3.3. Second-order constraints

Second-order constraints arise from simplifying hypotheses made with respect to camera motion and camera intrinsic parameters. Here, the camera roll angle is assumed to be zero. In other words, the vanishing line corresponding to the horizontal lines in the 3D world *is* horizontal in the image.
Expressing the fact that the two vanishing points have the same u coordinate amounts to writing:

$$(\mathbf{H_i^m})_{11}(\mathbf{H_i^m})_{32} = (\mathbf{H_i^m})_{12}(\mathbf{H_i^m})_{31}. \tag{5}$$

To add this constraint, we rewrite the fundamental Equation 4 as:

$$\begin{cases} \min \|\mathbf{A} h_i^m\|^2 \\ (h_i^m)^T \mathbf{B} h_i^m = 0, \end{cases} \tag{6}$$

where $\mathbf{B}$ is a symmetric matrix easily deduced from Eq. 4. This is a minimization problem under constraints, which can be solved by any standard method.

4. ESTABLISHING CORRESPONDENCES IN THE SPORT CONTEXT

In the case of soccer, we can “recalibrate” the homography whenever we have enough features to match. This should happen when the camera focuses on one of three zones of the game field, namely the two goal areas and the central area. However, the features that are used are very different for each of these types of areas: (1) lines only in the goal areas; (2) a circle and two lines in the central area.

4.1. Case 1: Sets of orthogonal lines

The features of interest are straight lines. To make the matching process easier, we have chosen to strongly rely on vanishing point information. This means that we try to segment all lines found in the image into 3 groups: 2 groups corresponding to the line segments in the “horizontal” directions (of the field plane) and one group for those that do not fall into either of these. Fig. 4 illustrates the notion of vanishing points.

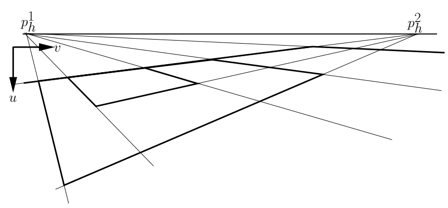


Figure 4: *Concept of vanishing points with conventions and notation. Heavy lines correspond to hypothetical lines on the ball field.*

4.1.1. Extraction of straight line segments

Our approach to line segment extration is based on a traditional method. The image gradient is computed with Canny’s edge detector: The result is a binary image from which we extract contour chains. A polygonal approximation of each chain is then recursively computed. This completes the segmentation into straight line segments. Significant improvements of the results are observed when color information is used for keeping or reject segments.
Exprimental results are shown in Fig 5. We see that most white straight lines in the goal area are correctly extracted. However, there are a few artifacts as well.

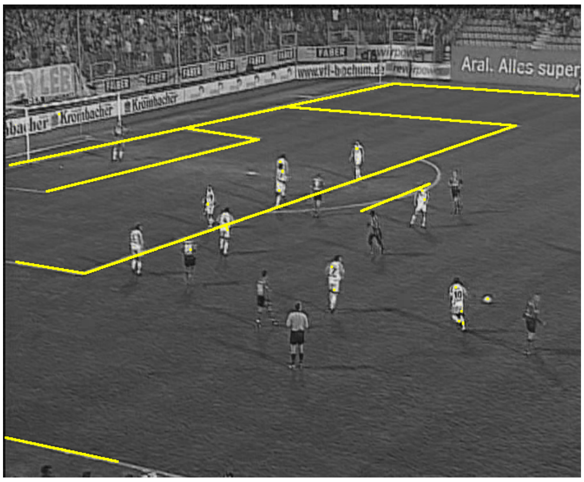


Figure 5: *Result of line-extraction process. Extracted lines are overlaid in yellow.*

4.1.2. Determination of vanishing points

It is well-known that straight-line grouping is essential for the efficient detection of vanishing points [6]. Grouping speeds up the processing of finding the vanishing points by reducing the search to a much smaller area of the parameter space.

Figure 4 illustrates the concept of vanishing point and shows our conventions and notation. As indicated earlier, since the roll angle is null in the case of TV cameras, the vanishing line is horizontal. Therefore, the two vanishing points $p_h^1 = (u_h, v_h^1, 1)$ and $p_h^2 = (u_h, v_h^2, 1)$ corresponding to the main orthogonal (horizontal) directions of the game field plane must lie on a horizontal line $u = u_h$. In addition, we suppose that the camera tilt and zoom are not varying too much, i.e. we have a rough knowledge of u_h, so that u_h obeys a constraint of the form:

$$u_{min} < u_h < 0.$$

We typically choose u_{min} equal to $-2d_u$, where d_u is the height of the image in pixels.

Our strategy for grouping the extracted line segments relies on assuming a very rough approximation of the vanishing line and on using this approximation to perform a selection among the edge segments. For all segments s_k detected in the image, we compute the intersection with the rough vanishing line $\hat{l}_k$, as in Fig. 6. Each intersection is characterized by a single scalar, i.e., angle θ_k, as defined in Fig. 6.

Then, we attempt to identify 2 clusters of angles in the θ_k histogram, that hopefully correspond to the main directions. Such clusters are illustrated in Fig. 7; they are clearly visible in the histogram. The third peak is linked to vertical directions, i.e., orthogonal to the field (e.g., goal posts).

Last, among each segmented group, a RANSAC [7] technique is applied to determine more precisely and more robustly the position of the vanishing points.

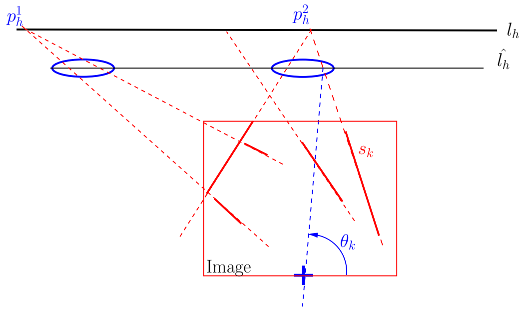


Figure 6: *Strategy for grouping segments*

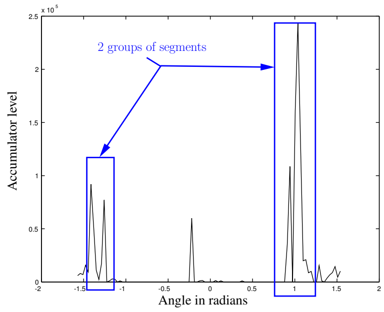


Figure 7: *Segment grouping: an example*

4.1.3. Matching segments lines to the model

Once vanishing points have been determined, the problem of identifying lines belonging to the model becomes much simpler as a segment either belongs to the pencil vanishing at p_h^1 or to the pencil vanishing at p_h^2.

We can state the problem as follows: Given two sets of *ordered* K (resp. L) lines passing through p_h^1 (resp. p_h^2), match the two sets to the corresponding sets in the model. The order is set on the basis of the order of the lines towards their respective vanishing points.

To solve this problem, we divide it into smaller subproblems: We first search for the *column* with the maximum number of characterizable intersections. Suppose this is column $\hat{k}$.

Our algorithm uses a local measure δ between points (in the model and in the image) to match sequences of crossings within columns. A column similarity measure is computed and is minimized over model columns. This measure is made robust to outliers (when a segment is missing or when there is false detection as in Fig. 8) by tolerating

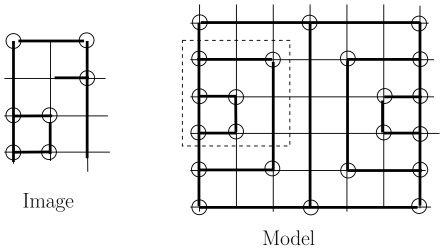


Figure 8: *Recognition of the lines pattern*

up to one consecutive differences in the sequences. In our case, δ is basically a discrete matching test between the forms of the line crossings ("T" or "+" crossings).
The model column which is the most "similar" to this one is selected to produce the first correspondences. Then the other columns are matched incrementally starting from the first matched column and using the point-to-point constraints generated in previous columns.

4.2. Case 2: Circular areas

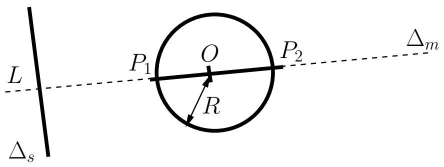


Figure 9: *Configuration of the central area (in the model).*

To keep our system as generic as possible, we make only one assumption regarding circular areas: it consists of a circle with a straight line Δ_m passing through its center O (see Fig. 9). We also suppose that another line (Δ_s) of the model, orthogonal to Δ_m, is visible. This is the case in soccer and in handball.
The intersections of the circle with Δ_m are denoted by P_1 and P_2, whereas L is the intersection between Δ_m and Δ_s. The projections in the image of O, L, P_1, and P_2 are points o, l, p_1, and p_2. The scalar R is the circle radius, so that the circle equation is

$$P^T \mathbf{E} P = 0, \tag{7}$$

where

$$\mathbf{E} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -R^2 \end{pmatrix}.$$

4.2.1. Ellipse detection and fitting

Most common approaches are based on accumulation in the parameter space, e.g., via the Generalized Hough Transform [8]. However, this method may be time-consuming. As a result, we exploit the particularities of our problem, i.e., color information and geometrical configuration (as in Fig. 9). Furthermore, we favor a RANSAC-like approach as opposed to an accumulation-based approach.

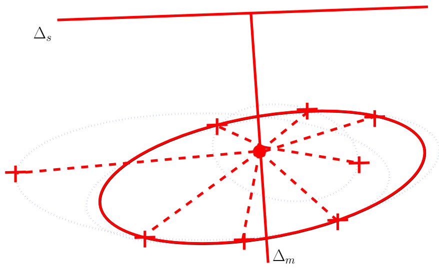


Figure 10: *A RANSAC-like approach for ellipse fitting.*

The idea is to detect the middle line Δ_m and lateral line Δ_s. Then we select on Δ_m one (or more, by sampling Δ_m) rough estimate(s) of a point inside the circle projection from a priori knowledge on the scene. Then we cast rays from this point and locate pixels where color profiles are close to those of a white line on a green background. This is the same approach as for tracking, as described in Section 5. The ellipse is then found among those correlation maximum points.

The algorithm we use is as follows:

```
detect middle line Δm;
detect lateral line Δs;
choose a rough estimate of ellipse center o;
S ← ∅
for all sampled rays around o do
    S ← points along rays with color transition;
endFor
while error superior to a treshold do
    randomly choose n points in S;
    error ← error in fitting ellipse to these points
end
refine ellipse equation with additional points;
```

Algorithm 1: Ellipse detection

A method for fitting an ellipse to a set of potentially interesting points is described elsewhere [9]. It is basically a total least-square approach on the conic parameters, under the constraint that these parameters are those of an ellipse. However, it is not robust to outliers: that is why we developped Algorithm 1, illustrated by Fig. 10. Our RANSAC approach is robust to interferences such as those due to players (as in Fig. 11) and inherits the precision of the ellipse fitting algorithm; it consists basically in choosing randomly a quorum of n of the initial points (e.g., 50%) to fit an ellipse.

Figure 11: *Ellipse detection and fitting.*

4.2.2. *Computing the homography from an ellipse*

We describe how to recover the homography $\mathbf{H_i^m}$ from the knowledge of the image configuration described in Fig. 9. Let $\mathbf{e}$ be the matrix describing the image ellipse, such that $p^T\mathbf{e}p = 0$ is the equation of this ellipse. The projection of the circle center is found in o according to a cross-ratio criterion.
Indeed, we suppose that cross-ratio $\{l, p_1, o, p_2\}$ is known, so that the position of o is easily recovered. We recall that the cross-ratio of four aligned, *ordered* points a, b, c, d is a *projective invariant*, that can be chosen as:

$$\{a, b, c, d\} = \frac{x_d - x_a}{x_d - x_b}\frac{x_c - x_b}{x_c - x_a},$$

where x_p is the coordinate of some point p along the straight line the points a, b, c, and d all belong to. In our case, $\{l, p_1, o, p_2\}$ (computed from the image) must be equal to $\{L, P_1, O, P_2\}$ (computed from the model). Once l, p_1 and p_2 are known from the previous step (ellipse detection), we have immediate access to o along line (p_1p_2). This approach is very precise: the errors between our estimate of the ellipse center o and the true center when it is visible in the image were found to be less than 1 pixel.
Let us denote the successive columns of $\mathbf{H_i^m}$ by h_1, h_2, and h_3. Below, we successively find useful expressions for the vectors h_1, h_2, and h_3.
To find h_3, we exploit the fact that point O (Fig. 9) is taken at the center of the reference frame, so that we can easily write:

$$h_3 = \mathbf{H_i^m}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \lambda o. \quad (8)$$

Let us choose λ so that $\mathbf{e} = (\mathbf{H_i^m})^T\mathbf{E}\mathbf{H_i^m}$. Based on Eq. 7, we have $O^TO = R^2$, so we must satisfy $h_3^T\mathbf{e}h_3 = R^2$. This leads to $\lambda = \frac{R}{\sqrt{o^T\mathbf{e}o}}$ and solves Eq. 8.

To find h_1, let us express it as:

$$h_1 = \mathbf{H_i^m}\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{R}\mathbf{H_i^m}\begin{pmatrix} R \\ 0 \\ 1 \end{pmatrix} - \frac{1}{R}\mathbf{H_i^m}\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

We find:

$$h_1 = \frac{1}{R}(\mu p_1 - h_3), \quad (9)$$

where μ is an unknown scalar factor.
From Eq. 7, we have $h_1^T\mathbf{e}h_3 = 0$, which implies via Equation 9,

$$\mu = R\frac{\sqrt{o^T\mathbf{e}o}}{o^T\mathbf{e}p_1}. \quad (10)$$

Finally, to find h_2, we make use of the following facts. First, from Eq. 7, we can immediately write

$$h_1^T\mathbf{e}h_2 = h_3^T\mathbf{e}h_2 = 0.$$

Second, from the fact that l is a line in the direction of $(0, 1, 0)^T$, we have $l^Th_2 = 0$.
These equations give us a linear system in the components of h_2, which can be immediately solved. Given that we have found h_1, h_2, and h_3, we have the desired $\mathbf{H_i^m}$ matrix.

5. HOMOGRAPHY FROM IMAGE-TO-IMAGE MOTION

Image-to-image motion refers to the apparent motion of features from frame to frame. Our goal is to track these features along sequences of images. The features we consider here are lines.

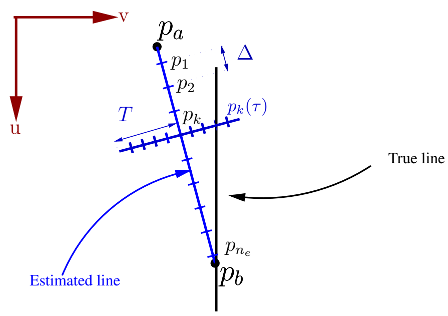
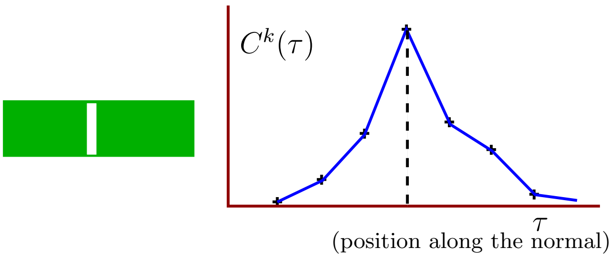


Figure 12: *Tracking lines from control points.*

Line tracking relies on the use of control points on each of the predicted segments from the model. Such an approach has proved successful in edge-based tracking [10]. Figure 12 illustrates the algorithm. Starting from an estimated prediction of a line segment $[p_a, p_b]$, a set of n_e

frame 68 frame 370 frame 690

Figure 14: *Example of line tracking.*



(a) (b)

Figure 13: *(a) Line color model. (b) Color pixel cross-correlation.*

control points p_k is built by regularly sampling this segment.
Then, at each p_k, along the normal to $[p_a, p_b]$, the edge point corresponding to the white-line is searched by correlation with a "typical" white line profile. Figure 13(a) shows the profile we use and typical correlation scores results are presented in Fig. 13(b).
Fig. 14 shows how the above algorithm performs on real images from a TV soccer sequence of approximately 700 frames. White points indicate the positions of the correlation maxima, and white lines the reconstructed segments after RANSAC. In this example, lines are lost only when they disappear from the image.
Finally, the tracked lines may be used in at least two ways. If at least four lines are being tracked, the model-to-image homography can be directly updated with Eq. 3. Otherwise, more features (typically points) are required to compute an image-to-image homography and update $\mathbf{H}_\mathrm{i}^\mathrm{m}$ according to Eq. 1.

6. RESULTS AND DISCUSSION

Experimental results of rectification are illustrated in Fig. 15. They are obtained by assembling the results of single rectifications in each of the goal zones and in the central area.
The fact that all 12 reprojections of the center circle and center line are almost perfectly aligned indicates that the algorithm performs well. Another indication is the fact that right angles are preserved in each of the goal areas.
Our experimental results indicate that our system gives a maximum error of about 5 pixels, which translate into localization errors on the order of a few tens of centimeters. These errors are doubtlessly the results of radial distortion, which are not taken into account for the moment, i.e., we suppose that our perspective model is perfect. Their effects on "rectified" images is clearly visible on top of the images.

7. CONCLUSIONS

This article presents some preliminary results of a full system we are developing to compute the homography (i.e., a matrix) between TV images of soccer games and a model of the soccer field. A significant feature of our approach is that it is fully autonomous. In other words, there is no need to install position and orientation sensors on TV cameras or position sensors on the players and the ball. This new technology shows great promise for "augmented" TV broadcasting.
Our current work is two-fold. On one hand, we want to design algorithms to take radial distortion into account simultaneously with homography estimation [11], as wide-angle cameras induce stronger distortion. On the other hand, we focus on strategies to achieve frame-rate performance, which requires improved point tracking algorithms.

8. ACKNOWLEGDEMENTS

The authors wish to thank Nicolas Gofflot and Pierre Gabriel for their precious collaboration. They also wish to express their gratitude to the Région Wallonne for providing financial support for this work via the WIST/DGTRE project

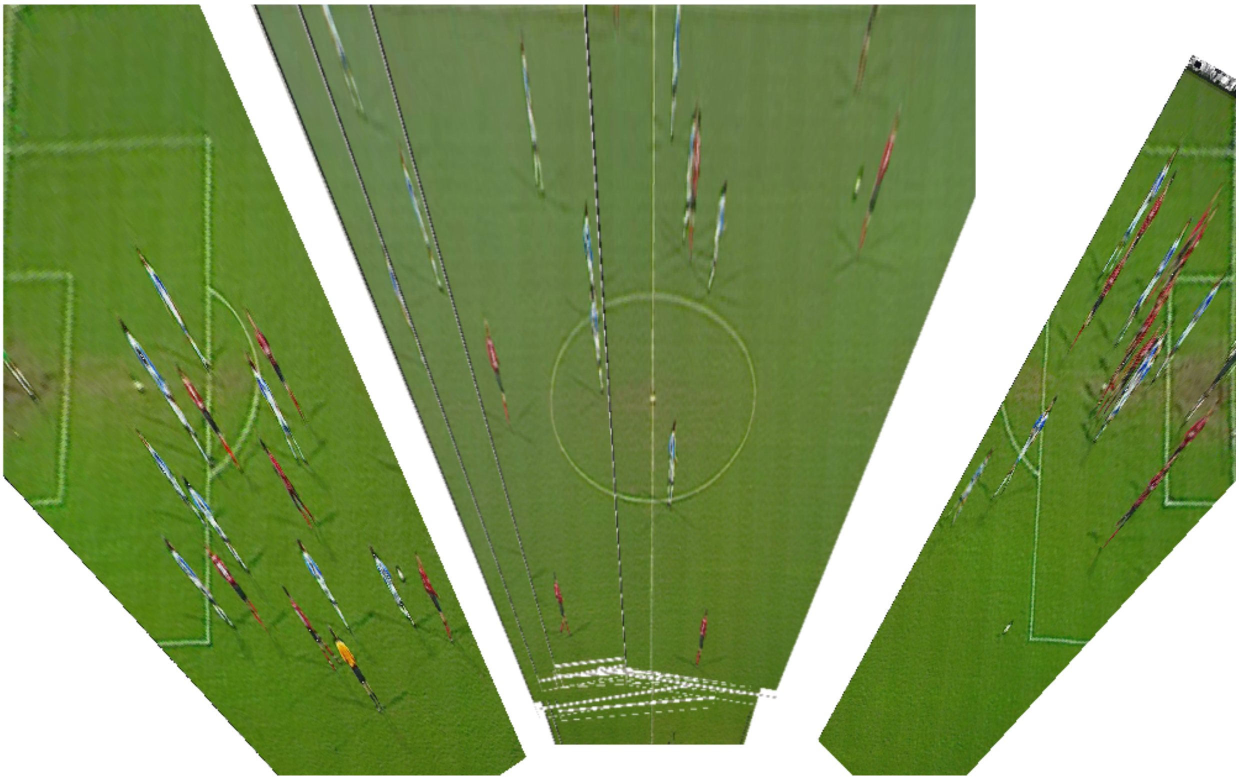


Figure 15: *Rectification of the soccer field.*

TRICTRAC (Convention 031/5439).

9. REFERENCES

[1] C. J. Needham, *Tracking and Modelling of Team Game Interactions*, Ph.D. thesis, The University of Leeds, October 2003.

[2] O. Faugeras, Q.-T. Luong, and T. Papadopoulo, *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*, MIT Press, 2001.

[3] D. Farin, S. Krabbe, P.H.N. de With, and W. Effelsberg, "Robust camera calibration for sport videos using court models," *SPIE Electronic Imaging*, 2004.

[4] A. Yamada, Y. Shirai, and J. Miura, "Tracking players and a ball in video image sequence and estimating camera parameters for 3D interpretation of soccer games," in *Proc. of IEEE International Conference on Pattern Recognition*, 2002.

[5] K. Okuma, Little J. J., and D. G. Lowe, "Automatic rectification of long image sequences," in *Proc. of the Asian Conf. on Computer Vision (ACCV'04)*, Jeju Island, Korea, Jan. 2004.

[6] W. Zhang and J. Kosecka, "Efficient detection of vanishing points," in *Proc. of the IEEE International Conf. on Robotics and Automation (ICRA'02)*, Washington DC, May 2002.

[7] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Comm. of the ACM*, vol. 24, pp. 381–395, 1981.

[8] Y. Xie and Q. Ji, "New efficient ellipse detection method," in *Proc. of IEEE International Conference on Pattern Recognition*, 2002.

[9] A. Fitzgibbon, M. Pilu, and R.Fisher, "Direct least-square fitting of ellipses," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, June 1999.

[10] T. Drummond and R. Cipolla, "Real-time tracking of complex structures for visual servoing," in *Workshop on Vision Algorithms*, 1999.

[11] A. Fitzgibbon, "Simultaneous linear estimation of multiple view geometry and lens distortion," in *Proc. of IEEE Computer Vision and pattern Recognition*, 2001.