Proceedings of the British Machine Vision Conference 2004, pp. 687–696

Robust incremental rectification of sport video sequences

Jean-Bernard Hayet Justus Piater
Jacques Verly
Department of Electrical Engineering and Computer Science (Institut Montefiore)
University of Liège
Building B28, B-4000 Liège, Belgium
FirstName.LastName@ulg.ac.be

Abstract

We describe an important element of an automatic sport analysis system. This element continuously estimates the image-to-model homography from the video stream of a single camera. Here, we focus on the incremental image-to-image updating of the homography matrix, which greatly facilitates real-time operation. This updating relies on the automatic tracking of interest lines and points and on their use for robust homography estimation. Results on real video sequences are shown.

1 Introduction

Sport broadcasters are beginning to rely on computer vision techniques to enhance their video productions. Examples are the smart overlay of advertisement on the grass without obscuring the players and the automatic analysis of games. This requires static (e.g., length) and dynamic (e.g., speed) metrics from one or more images produced by one or more cameras. With metric information, it becomes possible to rectify images, i.e., to project them onto a model of the game field, thereby facilitating subsequent analysis.

Current commercial systems for automatic rectification are expensive. Indeed, most of them rely on sensors embedded in cameras. They measure all relevant camera parameters, such as orientation in space and zoom.

In this paper, we focus on a subsystem of a fully-automated sport analysis system we are developing. This subsystem allows us to maintain a continuous estimate of the homographic mapping [4] between a model of the field and images of it, without having recourse to any kind of sensor on the cameras and/or the players. To achieve this functionality, we use three complementary strategies.

The first strategy computes the homography from scratch. It is used initially and then periodically to reinitialize the system. The second strategy quickly computes the homography from a current estimate by maintaining image-to-model correspondences, thereby enabling real-time operation. The third strategy updates the homography incrementally from image-to-image correspondences. It is helpful when image-to-model matches are not sufficient. Drifts due to this last mode of operation can be compensated for by peri-

This work was sponsored by the Région Wallone under DGTRE/WIST contract 031/5439.

odic reinitialization as provided for by the first strategy. The incremental updating is the object of this paper.

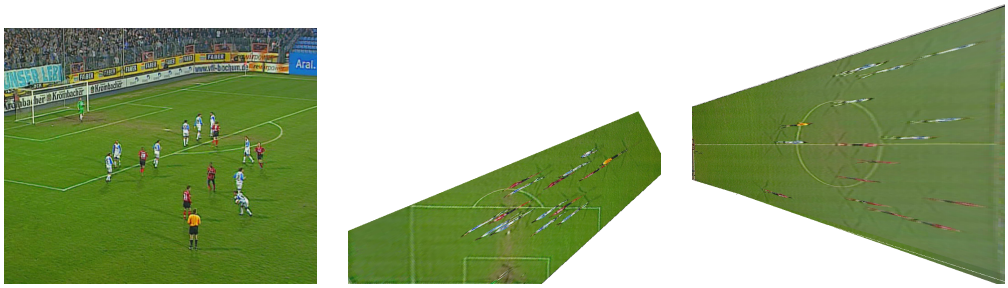
Other approaches have been investigated for estimating inter-image homographies in soccer sequences. Many of these approaches rely on lines only [3, 7], but this restricts their applicability to relatively narrow zones, mainly just around the goal zones. Of course, knowledge of image-to-image homographies also leads directly to rectification and mosaicing.

Our approach is based on tracking point features on and around the playing field. This approach is also used in [8], where the sport of hockey is considered. Using this technique in hockey is made easier by the fact that the game field is relatively small, with the result that several very distinctive features persist from image to image. By contrast, in the case of soccer, there are many instances where we have to rely on less well defined features such as patches of grass.

This paper is organized as follows. In Section 2, we briefly review the notion of homography and rectification. We also define notations, highlight important properties, and give an overview of our system. Sections 3 and 4 describe our algorithms for line and point tracking, respectively. Section 5 describes the estimation of the inter-image homographies. Section 6 shows experimental results obtained from real soccer TV sequences.

2 Homography and rectification

We consider homographies between an image (e.g., in a video stream) and a 2D model (e.g., of a soccer field). Once determined, a homography can be used to reproject, or rectify, the image so that it lines up with the model. This is illustrated in Fig. 1 in the case of the goals and central areas. Of course, the homography transformation is such that the rectified image is independent of all camera parameters, as it compensates for camera position, attitude, and internal parameters.



(a) (b) (c)

Figure 1: *Illustration of the role of homographic transformations. (a) Selected image in video stream. (b) Same image after rectification by homography. (c) Other image in the center-circle area after rectification.*

We represent each 2D image point p by its homogeneous coordinates, i.e., $(u,v,1)^T$. The model-to-image homography is represented by a 3×3 transformation matrix $\mathbf{H_i^m}$, where i stands for image and m for model. Matrix $\mathbf{H_i^m}$ can be related to the rigid transformation $(\mathbf{R_i}, t_i)$ linking the camera frame to the model frame. One can show that [4]

$$\mathbf{H_i^m} \sim \mathbf{K_i}[r_i^1 r_i^2 t_i], \tag{1}$$

where matrix $\mathbf{K_i}$ contains the camera internal parameters for image i and matrix $[r_i^1 r_i^2 t_i]$ is made up of the first two columns of the rotation matrix $\mathbf{R_i}$ followed by the column vector of translation. The 3×3 matrix $\mathbf{H_i^m}$ will often be represented by an equivalent 9×1 vector h_i^m, obtained by stacking the successive rows of $\mathbf{H_i^m}$.

The homography matrix $\mathbf{H_i^m}$ has 8 (i.e., $9-1$ because of the scaling factor) degrees of freedom (DOFs). Therefore, since each correspondence contributes 2 equations, a minimum of 4 point or line correspondences are necessary to estimate $\mathbf{H_i^m}$. It is significant that the required correspondences may come from a wide variety of features extracted from the imagery, such as points, lines, and circles.

Our goal is to have an estimate of $\mathbf{H_i^m}$ for any image in the sequence, even when we don't have enough correspondences. We manage to maintain such a continuous estimate of the homography by using 3 complementary strategies.

The first strategy exploits image-to-model correspondences. When a sufficient number of these correspondences are available, the homography can be estimated. 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 each application (such as the lines and circles of a soccer field). Feature matching implies the definition of feature characteristics and of comparison metrics. This operation is also likely to be time-consuming. Our system currently uses two kinds of features: lines and ellipses. It has robust means for detecting these features and matching them to the model, so that the image rectification can be computed as illustrated in Fig. 1.

The second strategy tracks features from image to image. Using the last computed homography, we can "project the model" into the current image. This allows us to focus the search for features on a much smaller area, close to the estimated feature positions, so that processing speed is increased.

The third strategy is useful to deal with situations where we have too few tracked correspondences to apply the previous strategy. Here, we estimate the homography between two successive images. In other words, we update the model-to-image homography incrementally. Let us suppose that $\mathbf{H_{i_0}^m}$ is the model-to-image homography corresponding to i_0. For a later image i, we compute the inter-image homography $\mathbf{H_i^{i_0}}$ that maps points from image i_0 to image i, so that

$$\mathbf{H_i^m} = \mathbf{H_i^{i_0}} \mathbf{H_{i_0}^m}. \tag{2}$$

To compute the estimate of $\mathbf{H_i^{i_0}}$, we use correspondences between two kinds of features, i.e., lines and points. The next two sections explain how we track these features from one image to the next.

3 Tracking lines

The first type of feature we track for computing homographies are line features corresponding to projections of lines of the model. Thus, in the case of ball games, such as soccer, there are very few trackable lines. However, they are very reliable.

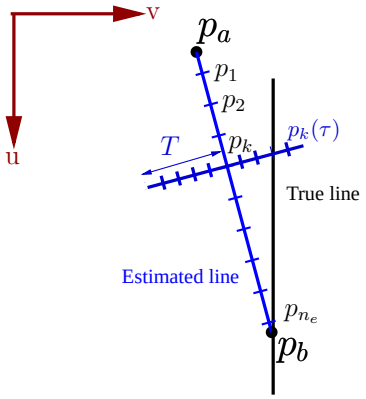


Figure 2: *Strategy for tracking lines using control points.*

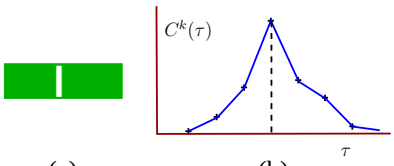


Figure 3: *(a) Color profile of line in model. (b) Cross-correlation scores.*

Tracking a line from image i to image $i+1$ involves using control points on the estimated position of the line in image i. Such approach has proved successful in edge-based tracking [2]. Figure 2 describes the procedure. Starting from a prediction $[p_a, p_b]$ of a true projected line segment, a set of n_e control points p_k is built by regularly sampling this segment.

Then, along the normal to $[p_a, p_b]$ going through each p_k, the edge point corresponding to the white-line is searched by correlation with a "typical" white line profile in a $[-T, T]$ interval. Figure 3(a) shows the profile we use. Figure 3(b) shows typical cross-correlation scores.

image 68 image 370 image 690

Figure 4: *Example of line tracking.*

Figure 4 shows how the above algorithm performs on real images from a TV soccer sequence of approximately 700 images. White points indicate the positions of the correlation maxima, and white lines the reconstructed segments after RANSAC [9]. In this example, lines are lost only when they disappear from the image.

Homographies can be computed incrementally, according to Eq. 2, by tracking at least 4 lines from image to image. They can also be computed by tracking a combination of a lesser number of lines and of other features, typically points, e.g., 2 lines and at least 2 points.

4 Tracking points

The second type of feature we track for computing homographies are points that have salient local texture characteristics. In ball games, these interest points can be found everywhere, including on the grass field, on advertisement banners, and in the spectator areas. For tracking these points, we use the Kanade-Lucas-Tomasi (KLT) tracker, and, more specifically, a popular pyramidal implementation of it [1].

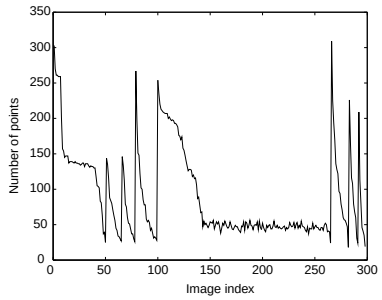


Figure 5: *Number of points available for computing the homography as a function of the image index.*

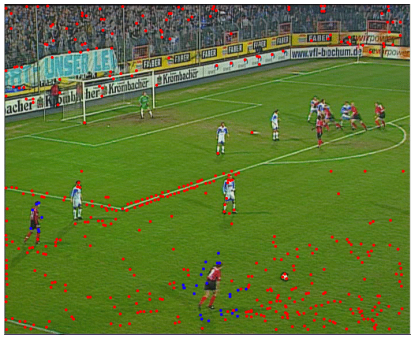


Figure 6: *Illustration of points used for tracking.*

The most suitable points for use with the KLT tracker have corner-like appearance, which translates into auto-correlation matrices with large eigenvalues. This led us to initialize our KLT tracker with the points given by a Harris point detector [6].

Of course, specific points cannot be tracked successfully over long sequences. They may indeed be lost due to occlusion or camera motion. Thus, we replenish the pool of tracked points whenever their number falls below a minimum value (set to 40 in our experiments). Figure 5 gives an illustration of this process over a long sequence of 300 images. In this example, the average lifetime for a set of Harris points is close to 10 images. Of course, this strategy can be efficiently improved, so that real-time operation can be achieved.

Examples of points tracked in a real sequence are shown in Fig. 6. We see that most points are found outside the field, e.g., among spectators. However, a reasonable number of points are also found on the field. These points generally correspond to "accidental structures" that are visible on the grass and that behave as interest points. However, some of these points also fall on players (which move and stick out of the field plane) and their shadows (which also move). Therefore, the procedure for estimating the homography based on such points must be robust enough to reject outliers, i.e., points that are moving and/or not in the plane of the field.

5 Computing the image-to-image homographies

Consider the projections of the same planar scene in 2 images i_1 and i_2 taken by the same camera from 2 vantage points characterized by a relative translation vector $t_{i_2}^{i_1}$ and

a relative rotation matrix $\mathbf{R}_{\mathbf{i_2}}^{\mathbf{i_1}}$. Then, the homography matrix relating these projections is given by [4]

$$\mathbf{H}_{\mathbf{i_2}}^{\mathbf{i_1}} \sim \mathbf{K}_{\mathbf{i_2}}(\mathbf{R}_{\mathbf{i_2}}^{\mathbf{i_1}} - \frac{1}{d_{i_1}} t_{i_2}^{i_1} n_{i_1}^T)\mathbf{K}_{i_1}^{-1}, \tag{3}$$

where n_{i_1} is the normal to the plane of the scene and d_{i_1} is the distance from the camera to the plane, both specified in the camera frame corresponding to i_1.

An estimate of $\mathbf{H}_{\mathbf{i_2}}^{\mathbf{i_1}}$ could theoretically be computed with a minimum of 4 correspondences. However, given the nature of the points we are tracking, we cannot say a priori which of these points are outliers, i.e., moving and/or not in the plane of the scene. Thus, we have to rely on robust estimators. RANSAC techniques have proven useful in such situations [9].

In our approach, we repeatedly compute estimates of $\mathbf{H}_{\mathbf{i_2}}^{\mathbf{i_1}}$ from randomly selected quadruples of points. Moreover, when searching in $\mathbb{R}^8$ (corresponding to the eight DOFs), local minima are likely to appear. Therefore, it is useful to try to reduce the dimensionality of the search space. This is achieved by making some simplifying assumptions.

The first assumption is that the camera roll angle is zero. This leads to

$$\mathbf{R} = \begin{pmatrix} \cos\theta_l & 0 & \sin\theta_l \\ -\sin\theta_t \sin\theta_l & \cos\theta_t & \sin\theta_t \cos\theta_l \\ -\cos\theta_t \sin\theta_l & -\sin\theta_t & \cos\theta_t \cos\theta_l \end{pmatrix},$$

where θ_l and θ_t are the pan and tilt angles, respectively. We also make a small-angle assumption, so that

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & \theta_l \\ 0 & 1 & \theta_t \\ -\theta_l & -\theta_t & 1 \end{pmatrix}.$$

The infinity homography [4] can then be written

$$\mathbf{H}_\infty = \mathbf{K}_{\mathbf{i_2}} \begin{pmatrix} 1 & 0 & \theta_l \\ 0 & 1 & \theta_t \\ -\theta_l & -\theta_t & 1 \end{pmatrix} {\mathbf{K}_{\mathbf{i_1}}}^{-1}. \tag{4}$$

We assume that u_0 and v_0 remain constant throughout the sequence and that the pixel aspect ratio is 1, so that we can write, for any image i, with coordinates centered on (u_0, v_0),

$$\mathbf{K}_\mathbf{i} = \begin{pmatrix} \alpha_i & 0 & 0 \\ 0 & \alpha_i & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5}$$

Substituting Eq. 5 into Eq. 4, we find

$$\mathbf{H}_\infty = \begin{pmatrix} \frac{\alpha_{i_1}}{\alpha_{i_2}} & 0 & \alpha_{i_1}\theta_l \\ 0 & \frac{\alpha_{i_1}}{\alpha_{i_2}} & \alpha_{i_1}\theta_t \\ -\frac{\theta_l}{\alpha_{i_2}} & -\frac{\theta_t}{\alpha_{i_2}} & 1 \end{pmatrix}. \tag{6}$$

Finally, we assume that the inter-image homography is equal to the infinity homography, i.e., $\mathbf{H}_{\mathbf{i_2}}^{\mathbf{i_1}} \approx \mathbf{H}_\infty$. This is reasonable since the points we track are far away, i.e., at infinity

for all practical purposes. Moreover, TV cameras are generally stationary, so that the translation component can be neglected.

Let us denote by $\{(p^k_{i_1} = (u^k_{i_1}, v^k_{i_1}), p^k_{i_2} = (u^k_{i_2}, v^k_{i_2}))\}_{1 \leq k \leq K}$ the K pairs of points that have been successfully tracked and matched between the two images i_1 and i_2. For convenience, we will use the notation K even if this number depends on i_1 and i_2.

Considering Eq. 4, and aiming at using a linear formulation for fast estimation, we can state the problem as in the model-to-image case:

$$\min_{h \in \mathbb{R}^6,} \|\mathbf{B}h\|, \tag{7}$$

where $\mathbf{B}$ is a $2K \times 6$ matrix that contains all the pairing data

$$\mathbf{B} = \begin{pmatrix} u^1_{i_1} - u_0 & 1 & 0 & -(u^1_{i_1} - u_0)(u^1_{i_2} - u_0) & -(v^1_{i_1} - v_0)(u^1_{i_2} - u_0) & -(u^1_{i_2} - u_0) \\ v^1_{i_1} - v_0 & 0 & 1 & -(u^1_{i_1} - u_0)(v^1_{i_2} - v_0) & -(v^1_{i_1} - v_0)(v^1_{i_2} - v_0) & -(v^1_{i_2} - v_0) \\ & & \cdots & & & \\ u^K_{i_1} - u_0 & 1 & 0 & -(u^K_{i_1} - u_0)(u^K_{i_2} - u_0) & -(v^K_{i_1} - v_0)(u^K_{i_2} - u_0) & -(u^K_{i_2} - u_0) \\ v^K_{i_1} - v_0 & 0 & 1 & -(u^K_{i_1} - u_0)(v^K_{i_2} - v_0) & -(v^K_{i_1} - v_0)(v^K_{i_2} - v_0) & -(v^K_{i_2} - v_0) \end{pmatrix}.$$

The over-determined system corresponding to Eq. 7 is robustly solved by RANSAC by using samples of only 3 points. To ensure that the domain of validity of the computed homography is as large as possible, we compute, for each sample, an empirical score that depends (1) on the number of points $p^k_{i_2}$ lying "close" enough to their counterparts $p^k_{i_1}$ after transformation by the current estimate of the homography matrix, and (2) on the spatial extent of the set of these best matching pairs.

Then, the final solution h is given by the eigenvector of $\mathbf{B}^T\mathbf{B}$ corresponding to the smallest eigenvalue, where $\mathbf{B}$ is build from the subset of the K pairs corresponding to the best matching pairs.

Line pairings are introduced in a completely equivalent way, based on the fact that, if we have $\mathbf{H}^{\mathbf{i_1}}_{\mathbf{i_2}} p^k_{i_1} = p^k_{i_2}$ for points, we necessarily have $(\mathbf{H}^{\mathbf{i_1}}_{\mathbf{i_2}})^T l^k_{i_2} = l^k_{i_1}$ for lines.

6 Results

All of our tests were carried out on various parts of a 5-minute, highly-dynamic soccer sequence. In our experiments, we used approximately 300 points for computing inter-image homographies but, generally, only a small fraction of them, typically around 50, were accepted as inliers by RANSAC. They are depicted in red in the example of Fig. 7.

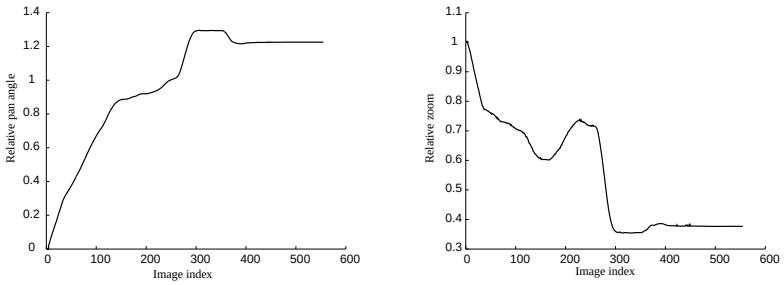
Figure 8 shows a mosaic built from a 200-images free-kick sequence. The rapid change in zoom (illustrated by Fig. 7) and the significant dynamics is a serious challenge for tracking, since image quality become very low in some images. However, the algorithm handles this part of the sequence well. Alignments are indeed precisely maintained.

Another challenging example is the "counter-attack" sequence of Figs. 12-13. This is a long sequence with fast camera rotation. Example images are shown in Figs. 10-12. Figure 13 shows the result of mosaicing. We overlayed (a) the reference image (in black), (b) two reference, horizontal lines (in yellow) to judge the alignment, and (c) the projections of four parallel lines that should meet at a vanishing point (in blue).

Figure 7: *Middle image in the "free kick" sequence*

Figure 8: *Mosaicing in a highly dynamic free-kick sequence.*

The excellent appearance of the mosaic of Fig. 13 shows that the various inter-image homographies were correctly estimated.



(a) (b)

Figure 9: *(a) Relative pan angle and (b) relative zoom, both as a function of image index for "counter-attack" sequence of Fig. 13*

The knowledge of the homography $\mathbf{H_i^{i_0}}$ for each image instant allows us to extract the value of important camera parameters from the general expression of $\mathbf{H_i^{i_0}}$ in the case of rotation without roll [7]. For example, one finds that the relative pan angle θ_l, relative tilt angle θ_t, and the relative zoom $\frac{\alpha_i}{\alpha_{i_0}}$ between images i_0 (reference) and i are given by

$$\theta_t = \arctan(-sgn(\mathbf{H_i^{i_0}})_{23})\sqrt{\frac{(\mathbf{H_i^{i_0}})_{22}(\mathbf{H_i^{i_0}})_{33}}{(\mathbf{H_i^{i_0}})_{23})(\mathbf{H_i^{i_0}})_{32})}} \quad (8)$$

$$\theta_l = \arccos\frac{(\mathbf{H_i^{i_0}})_{11}}{(\mathbf{H_i^{i_0}})_{22}}\cos\theta_t \quad (9)$$

$$\frac{\alpha_i}{\alpha_{i_0}} = \frac{(\mathbf{H_i^{i_0}})_{11}}{\cos\theta_l}. \quad (10)$$

Equations 9 and 10 are illustrated in Fig. 9. The variations of θ_l and $\frac{\alpha_i}{\alpha_{i_0}}$ in the graphs

are in good agreement with the camera pan and zoom parameters that can be qualitatively inferred from watching the sequence.

Our TV images clearly suffer from a significant radial distortion, which contributes to accumulating errors in the estimated homography. However, one should remember that a soccer field contains several distinctive zones, where the estimate of $\mathbf{H_i^m}$ can be recalibrated to the model, as is the case in mobile-robot navigation.

7 Conclusion

We have developed a robust approach for incrementally estimating the homographic transformation between a model and an image. The estimate of the transformation can be used for image rectification and mosaicing. Our image-to-image homography estimation technique is based on the use of interest points combined with straight lines. Points and lines are tracked and contribute to homography estimation through a RANSAC procedure.

The algorithm was successfully tested by mosaicing long video sequences, without recalibrating the homography to the model with line or circle features. Our current work involves the processing of radial distortion on the basis of existing methods [5] and the validation of the techniques over larger databases corresponding to various sports.

References

[1] J.Y. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker. Technical report, OpenCV documentation, 2000.

[2] T. Drummond and R. Cipolla. Real-time tracking of complex structures for visual servoing. In *Workshop on Vision Algorithms*, pages 69–84, 1999.

[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] 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.

[5] A. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2001.

[6] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference, Manchester*, pages 147–151, 1988.

[7] H. Kim and K. S. Hong. Robust image mosaicing of soccer videos using self-calibration and line tracking. *Pattern Analysis and Applications*, (4):9–19, 2001.

[8] 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, January 2004.

[9] C. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.

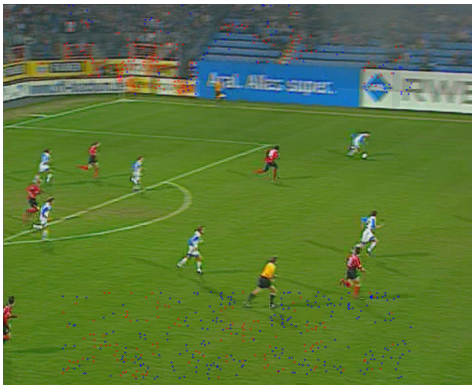


Figure 10: *Image 1025 of "counter-attack" sequence.*

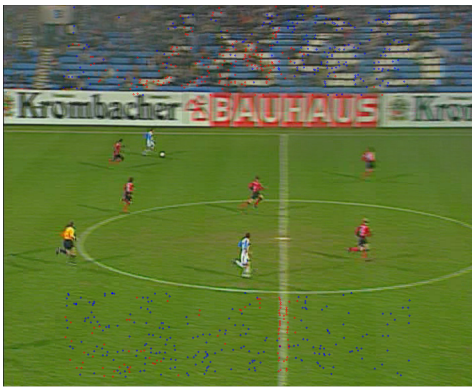


Figure 11: *Image 1100 of "counter-attack" sequence.*

Figure 12: *Image 1309 of "counter-attack" sequence.*

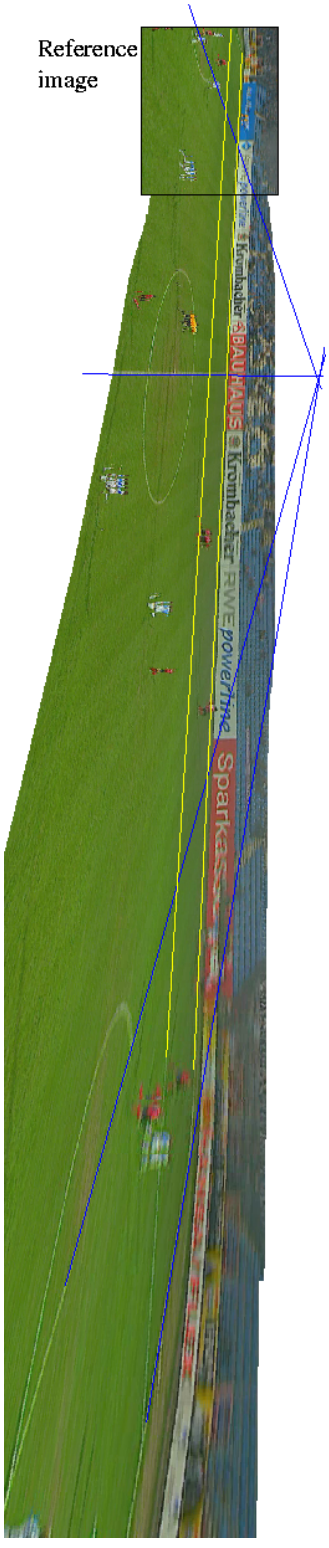


Figure 13: *Mosaicing for "counter-attack" sequence (from image 1010 to image 1420). Sequence is long and exhibits fast camera rotation.*