Automatic soccer field of play registration

Carlos Cuevas, Daniel Quilón, Narciso García

PII: S0031-3203(20)30083-2
DOI: https://doi.org/10.1016/j.patcog.2020.107278
Reference: PR 107278

To appear in: *Pattern Recognition*

Received date: 18 July 2019
Revised date: 30 December 2019
Accepted date: 12 February 2020

Highlights

- Automatic registration of soccer images on a model of the field of play.
- Field of play segmentation and detection of representative line segments.
- Probabilistic decision tree to classify the set of all straight lines.
- Center circle modelling using a Least Square Fitting algorithm for ellipses.
- Three-step validation stage to determine whether the registration is correct.

Automatic soccer field of play registration

Carlos Cuevas[a,*], Daniel Quilón[a], Narciso García[a]

[a] *Information Processing and Telecommunications Center (IPTC) and ETSI Telecomunicación, Universidad Politécnica de Madrid (UPM), 28040 Madrid, Spain*

Abstract

This paper proposes a strategy for the automatic registration of soccer images on a model of the field of play. First, a robust and efficient preprocessing is applied to discard the areas of the image that do not belong to the field of play and eliminate most edge points that are not part of the line marks. Then, a novel probabilistic decision tree is used to identify the most probable classification for the set of all the straight lines in the image. Additionally, the line surrounding the center circle is also modeled for providing results when only few straight lines are visible. Finally, a three-step analysis stage is applied to ensure the validity of the results.

To assess its quality, the strategy has been tested on several sequences corresponding to three stadiums with different characteristics. The results obtained have shown that the registration is successful in most images (94%).

Keywords: soccer, football, field of play, playing field, registration, segmentation, line segment detection, line classification, validation

1. Introduction

Currently, soccer is the most popular sport in the world [1], with more than 265 million players in more than 200 countries [2] and with the largest television audience. Thanks to this great popularity and to the technological advances produced in the last years, several artificial vision applications have been proposed

[*] Corresponding author
Email address: ccr@gti.ssr.upm.es (Carlos Cuevas)

to carry out the automatic analysis of soccer matches. In addition, these applications are increasingly demanded not only by the audience, but also referees, coaches, and players.

The applications for soccer video analysis cover a wide range of techniques with very varied uses, such as the detection and tracking of players [3] or the ball [4], the collection of match statistics (e.g., the distance covered by players, the speed of a player during a given move, etc.) [5], or the detection of events (e.g., off-side positions, goal scoring, etc.) [6].

Many of these applications provide successful results. However, they do not typically perform in real-time, which is an essential requirement so that they can be used in live transmissions. Additionally, they typically propose strategies that are focused on solving/analyzing very specific aspects, but they are not integrated with others so that they can be used as a commercial product suitable for TV networks or soccer clubs. For these reasons, new applications are continually being presented.

1.1. Contribution

We propose an efficient strategy for the automatic registration, on a model of the field of play, of images captured by the Master Camera[1] in soccer broadcasting on TV. Image registration is a necessary stage in several soccer video analysis applications, since it is required, for example, for collecting statistics (e.g., position of the players throughout the match), or with augmented reality purposes (i.e., including virtual objects on the images).

The specific contributions of this paper are:

1. A segment detection strategy is used to discard most unwanted edge data. This strategy not only allows an accurate detection of the line marks, but

[1] It is the one used most of the time in soccer broadcasting. It is placed approximately on the extension of the halfway line. It performs pan, tilt, and zoom movements, but not rotations around its longitudinal axis (i.e., no roll).

also prevents false detections due to the edges in the players or the ball. Additionally, it avoids getting duplicate detections.

2. The detected lines are not classified independently but using a probabilistic decision tree that identifies the most probable classification for the set of all detected lines. Thereby, the results of the classification are more robust than those obtained using the methods that classify the lines one by one. Moreover, lines not belonging to the field of play but to the goals or the billboards are successfully discarded.
3. The center circle of the field of play is modeled with an efficient strategy that combines a region growing algorithm and a Least Square Fitting (LSF) algorithm. In contrast to previous strategies, the proposed one is capable of obtaining very accurate results from a single intersection between the center circle and the halfway line, even if the center circle does not appear complete in the image.
4. A three-step validation stage is applied to determine whether the registration is correct. This is a significant feature of the strategy since, unlike previous registration strategies, we avoid providing incorrect results.

1.2. Organization

The paper is organized as follows. First, the related work is summarized in section 2. Then, an overview of the proposed strategy is provided in section 3. Sections 4 to 7 describe the different stages of the strategy. Finally, sections 8 and 9 contain, respectively, the obtained results and the conclusions. Additionally, the field of play model and its key-points are described in Appendix A.

2. Related work

Many strategies are available nowadays that, similarly to the proposed one, are aimed at the estimation of the homographies that allow to register images of soccer matches in a model of the field of play. In this section, we briefly summarize the most relevant ones.

In [7], a self-calibration method for non-fixed cameras is proposed, which is able to register the images by comparing the intersections of the detected line marks with a set of possible patterns. In [8], once the detected lines have been classified into longitudinal or transverse, the homography matrices are estimated by analyzing all the possible matches between the intersections of the lines and the key-points in the model. In [9], the registration is performed by establishing matches between the detected lines and the lines in the model. In [10, 11, 12], the sets of longitudinal and transverse lines are used to determine two vanishing points that are used to perform the registration.

Although these strategies provide successful results in many images, they fail when only few lines are visible, which is typical in the central part of the field of play. Besides, these strategies do not include any procedure to determine if the results are correct.

To register images with few visible lines, some strategies [13, 14] include algorithms to relate these images to images that have been previously registered. However, the quality of their results decreases drastically as the number of consecutive images with few lines increases.

Some strategies also include algorithms to register the images using data from the center circle of the field of play [15]. Some of them [16] use a 6-dimensional Hough transform (HT) to detect ellipses since, due to the perspective of the images, the center circle is seen as an ellipse in the images. The main drawback of these strategies is their high computational and memory costs [17, 18]. Consequently, some authors have proposed probabilistic methods focused on reducing such costs [17, 18]. However, since they do not consider all the significant edge points in the images, their accuracy is not high enough. Other strategies, also to improve the computational efficiency, try to model the ellipses starting from some previously known ellipse points [19, 20, 21]. However, their assumptions limit the amount of images in which they provide successful results. An additional limitation of HT-based strategies is that they are not able of obtaining successful results in images where the center circle does not appear complete. As an alternative to HT-based algorithms, some authors [22] have proposed strategies using LSF methods [23]. However, they are are very sensitive to the presence of outliers (i.e., points that do not belong to the ellipse) [24]. Others [25, 26] have proposed algorithms based on the detection of straight lines tangent to the ellipse [27], or methods looking for edge points that do not belong to the longitudinal or transverse groups of lines [12].

All these strategies include preprocessing steps that try to eliminate the image data that can hinder the registration process. The quality of the results obtained with these preprocessing stages greatly influences the quality of the final results.

One of the most typical preprocessing stages consists in discarding irrelevant areas (e.g., the stands) by segmenting the field of play. Typically, this is done by analyzing the H component of the HSV color space. Some works are based on applying fixed thresholds around the H value for green [28]. Whereas other works [29, 30] try to identify the dominant color of the field of play and then they infer thresholds around such dominant color. Because the H component is cyclically defined, both the selection of thresholds and the discrimination of statistical color modes are complex, which result in wrong results in images that alternate dark and light grass bands or contain shaded areas. Additionally, these strategies are very sensitive to the size of the histogram bins.

Another typical preprocessing step consists on detecting the image-points belonging to the line marks [13, 31, 32]. This detection is typically based on methods that apply the HT to edge images that are obtained with the Sobel [3, 33] or the LoG [34, 35] operators. These methods are computationally efficient and provide successful results in simple images. However, they fail in images containing several edge points that do not correspond to the line marks, but to the players, the ball, the goals, or the billboards. Moreover, some of them provide duplicate detections.

3. System overview

The proposed strategy comprises a preprocessing block and three processing stages, as shown Figure 1.

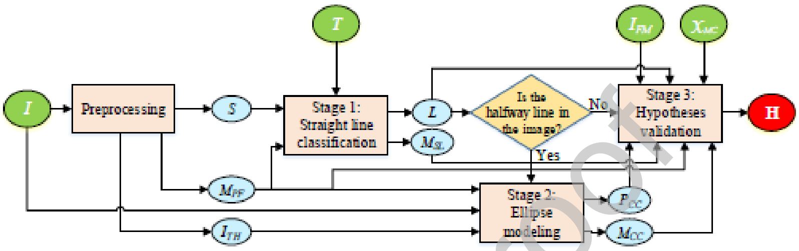

Figure 1: Block diagram of the proposed strategy. Round-edged blocks denote data, rectangular blocks denote processes, and the diamond block indicates decisions. The green and red blocks indicate, respectively, the inputs and the output.

In the preprocessing block (section 4), an analysis based on the green chromaticity is applied for segmenting the field of play. Additionally, the most representative line segments in the field of play are detected thanks to a strategy that allows discarding most unwanted edge data.

In Stage 1 (section 5) the straight line marks are detected. Then, a probabilistic decision tree is used for classifying the detected lines into 6 possible groups by matching their tilt with a set of tilt models.

Stage 2 (section 6) is focused on detecting and modeling the center circle of the field of play. This stage is only applied if the halfway line of the field of play is one of the lines resulting from the classification carried out in Stage 1.

Finally, in Stage 3 (section 7), the data from the previous modules are used to generate hypotheses that establish correspondences between image points and key-points in the model of the field of play. A three-step validation analysis is applied to determine if any of the considered hypothesis provides the correct registration.

4. Preprocessing

The segmentation of the field of play is a necessary initial stage in most soccer video analysis applications [36], since it allows to discard irrelevant areas, such as stands, (digital) billboards, etc.

The proposed strategy uses the segmentation method in [37], which is based on the estimation of the probability density function of the green chromaticity of the images, to obtain a binary mask, M_{PF}, corresponding to the segmentation of the field of play. This method provides high-quality results in images taken from different viewpoints and in complex situations (e.g., lawn striping resulting from bending the grass in different directions, or heavily shaded stadiums). Moreover, it does not require any user interaction. This chromaticity-based strategy has proven to exceed the quality of the results achieved by the state of the art strategies that, as stated in section 2, typically analyze the H component of the HSV color space. Figure 2 shows two segmentation results obtained in images from different stadiums. It can be observed that both results are very accurate.

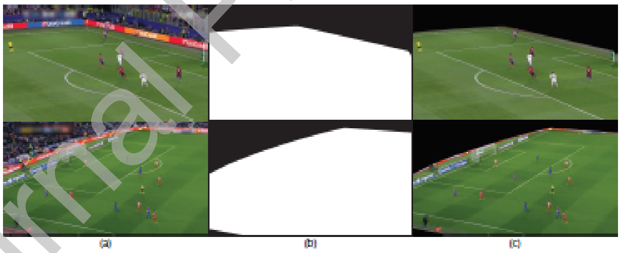

Figure 2: Segmentation results in two stadiums. (a) Original images. (b) Obtained binary masks, M_{PF}, (c) Segmented images.

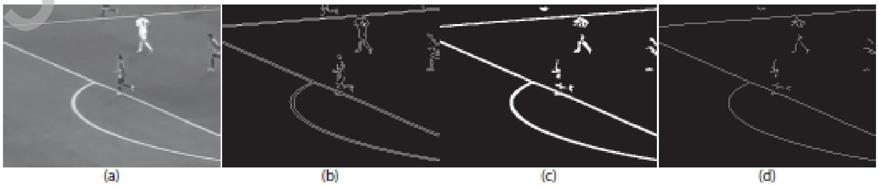

Figure 3: (a) Gray-scale image, I_g. (b) Edge image obtained with the Sobel operator. (c) Top-hat image, I_{TH}. (d) Image of segments from the ELLSF algorithm.

Additionally, a segment detection method to discard the vast majority of the unwanted edge data is included in this stage. This method improves the accuracy in the detection of the straight lines, avoiding typical errors due to the edges in the players or the ball.

Let I be the RGB image to process. First, a gray-scale image, I_{TH}, in which the brighter elements of I (e.g., the lines) are highlighted, is obtained by applying the top-hat transform as follows:

$$I_{TH} = I_g - I_g \circ e_{sd}, \tag{1}$$

where I_g is a gray-scale image obtained by masking the normalized luminance data of I with M_{PF}, $\circ$ denotes the opening morphological operation, and e_{sd} is a structuring element whose size must be somewhat greater than the width of the lines (in the performed experiments, a 7×7 structuring element has been used). Thus, in contrast to other strategies using the Sobel [33] or the LoG [35] operators, we avoid detecting duplicate edges (see Figure 3.b).

Then, the Edge Linking and Line Segment Fitting (ELLSF) algorithm [38] is applied on I_{TH} to detect and label line segments. This algorithm provides a list of connected edge points that allows constructing a mask of segments (see third and fourth images in Figure 3).

Finally, to remove the small but numerous segments that appear on the billboards, players, goals, etc., the marginal utility principle [39] is applied as follows. Let $S_0 = \{s_i\}_{i=1}^{n_s}$ be the set of n_s segments detected with the ELLSF algorithm, sorted according to their length, c_i (i.e., the amount of pixels corresponding to each segment), so $c_i \geq c_{i+1}$. Let us consider a sequence of n_s incremental subsets of segments, $S'(i) = \{s_j\}_{j=1}^{i}$ $i \in \{1, \ldots, n_s\}$, each subset containing the first i segments in S_0. Then, the utility of each subset is defined as

$$U\left(S'(i)\right) = \frac{1}{i}\sum_{j=1}^{i} c_j,\ i \in \{1, \ldots, n_s\}. \tag{2}$$

According to the marginal utility principle, the last useful segment in S_0, n_{ss}, is the first one satisfying that this utility function has decreased less than 1 unit with respect to its previous value. That is:

$$n_{ss} = \min \{\ i \in \{2, \ldots, n_s\}\ \mid\ U(S'(i-1)) - U(S'(i)) < 1\ \}. \tag{3}$$

Therefore, the final set of segments will consist only of these n_{ss} representative segments. That is, $S = S'(n_{ss}) = \{s_i\}_{i=1}^{n_{ss}}$.

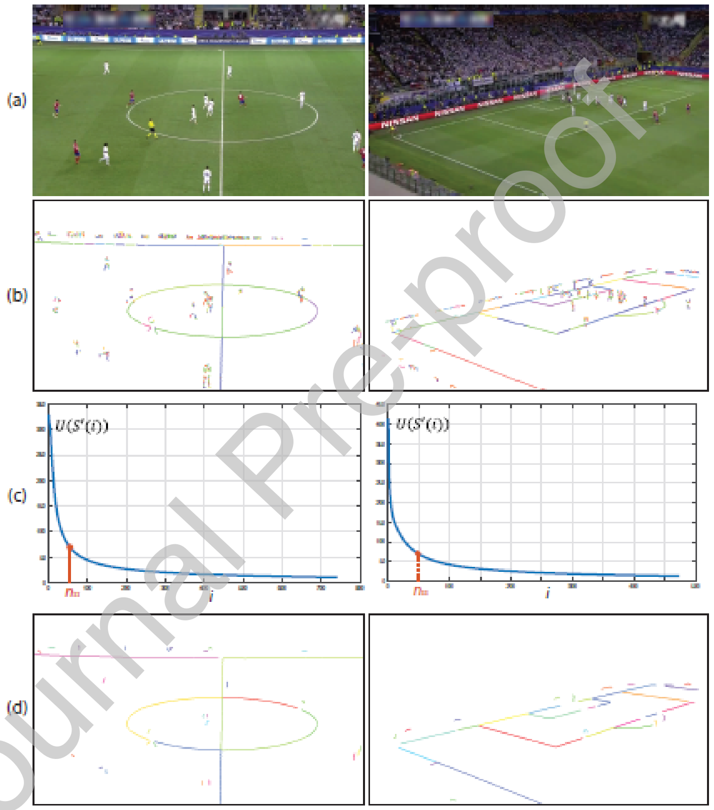

Figure 4: (a) Original images, I. (b) Initial sets of segments, S_0. (c) Utility functions, $U(S'(i))$, where the dotted lines mark the last useful segment, n_{ss}, in each set. (d) Sets of final segments, S. To facilitate its visualization, the segments have been represented in varied colors.

Figure 4 shows some results obtained with the proposed segment detection method. It can be observed that after applying the marginal utility principle, most spurious segments have been successfully removed.

5. Straight line classification

Once the most representative line segments on the field of play have been detected, they are used to detect (subsection 5.1) and classify (subsection 5.2) the visible straight lines of the field of play. This classification is one of the most representative contributions of this paper since, in contrast to state of the art strategies, which classify independently the lines, it includes a novel probabilistic decision tree to identify the most probable classification for the set of all detected lines.

5.1. Line detection

First, a binary mask, M_S, indicating the pixels belonging to the set of segments in S is constructed. Then, the HT [16] is applied on M_S. From the maxima of the HT, the set of the most representative $n_l^{'}$ straight lines in the image is obtained, $L^{'} = \{l_i\}_{i=1}^{n_l^{'}}$, where the i-th detected line is defined as $l_i = (\gamma_i, \theta_i)$, being γ_i the amount of pixels in M_S that have voted to such line in the HT, and $\theta_i \in [-\pi/2, \pi/2)$ the slope of the line to the vertical axis of the image (i.e., its tilt).

To facilitate the subsequent line classification step, the lines are added to $L^{'}$ according to the decreasing values of γ, either until the number of votes of the next line to be added is lower than the 10% of γ_1, or until reaching a maximum of $n_l^{'} = 10$ lines (since this is the maximum number of straight lines that can be seen on each of the halves of the field of play). Consequently, the lines in $L^{'}$ are ordered according to how reliable their detection has been.

Figure 5 shows some results obtained with the proposed detection strategy and the approaches in [33] and [35], in three images with different zoom levels.

The strategy in [33] uses the Sobel operator to obtain the edge images. Then, it applies the HT to detect the most significant lines. It can be observed that this strategy does not detect all the lines in any of the analyzed images. Moreover, it results in several false detections that, in the case of the image with the highest zoom, are due to duplicate lines.

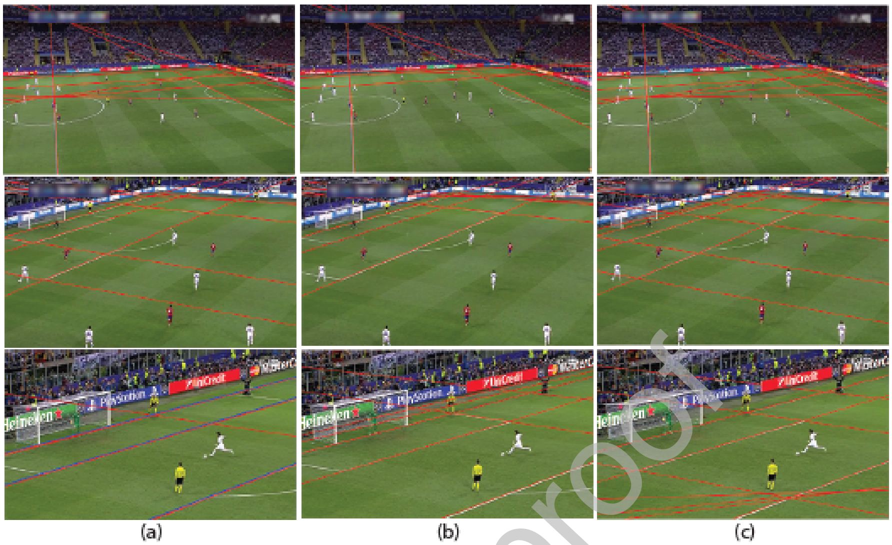

Figure 5: (a) Results with the strategy in [33]. (b) Results with the strategy in [35]. (c) Results with the proposed strategy. Color notation: Detected lines have been depicted in red, with the exception of duplicate detections, which have been depicted in blue.

The strategy in [35] applies the HT on edge images obtained with the LoG operator. In addition, it includes a step to avoid duplicate detections by discarding parallel and too close lines. Similarly to the previous one, it does not avoid misdetections and it also gives rise to false detections.

Regarding the proposed strategy, it can be observed that it is the only one that avoids misdetections in the three images. This is because most unwanted edge points (i.e., those that do not belong to line marks) are discarded with the segment detection performed in the preprocessing module. Additionally, since the edge data are obtained from the ELLSF algorithm combined with the Top-Hat transform, in contrast to other strategies, the proposed one avoids duplicate detections and provides line equations that are completely centered on the line marks. On the other hand, the proposed strategy also results in some false detections. However, such false detections will be easily discarded in the following stages.

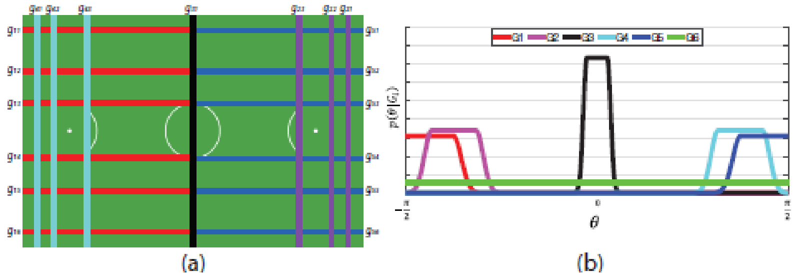

Figure 6: (a) Field of play model with line classification nomenclature. (b) Tilt models corresponding to the six considered groups of lines.

5.2. Line classification

According to the IFAB Laws of the Game [40], the field of play is divided into two halves by a halfway line. The lines in each of these halves can be parallel to the touchlines (longitudinal lines), or parallel to the goal lines (transverse lines). Taking into account these characteristics, the following five groups of lines, illustrated in Figure 6.a, have been considered:

- $G_1 = \{g_{1i}\}_{i=1}^{6}$: Longitudinal lines in the left side of the field of play, sorted from top to bottom.
- $G_2 = \{g_{2i}\}_{i=1}^{3}$: Transverse lines in the right side of the field of play, sorted from right to left.
- $G_3 = \{g_{31}\}$: Halfway line.
- $G_4 = \{g_{4i}\}_{i=1}^{3}$: Transverse lines in the left side of the field of play, sorted from left to right.
- $G_5 = \{g_{5i}\}_{i=1}^{6}$: Longitudinal lines in the right side of the field of play, sorted from top to bottom.

In addition, a sixth group, G_6, has also been considered to take into account the noisy detections (i.e., lines that do not belong to any of the five previous groups).

To classify the detected lines according to their tilt values, the set of tilt models $T = \{\mathrm{p}\,(\theta|G_j)\}_{j=1}^{6}$ has been defined , where $\mathrm{p}\,(\theta|G_j)$ is the pdf corresponding to the j-th group of lines. These models appear illustrated in Figure 6.b. The first five models have been constructed from sets of tilt values corresponding to manually annotated lines appearing in images captured in different stadiums (i.e., with different locations of the Master Camera) and with different camera pan/tilt configurations. These models are defined as:

$$\mathrm{p}\left(\theta|G_j\right) \propto \begin{cases} \exp\left(-\frac{(\theta-\theta_{j,1})^2}{2\sigma_j^2}\right), & \theta < \theta_{j,1} \\ 1, & \theta \in [\theta_{j,1}, \theta_{j,2}] \\ \exp\left(-\frac{(\theta-\theta_{j,2})^2}{2\sigma_j^2}\right), & \theta > \theta_{j,2} \end{cases} \tag{4}$$

where $(\theta_{j,1}, \theta_{j,2})$ are, respectively, the minimum and maximum tilt values in the set of lines corresponding to the j-th group, and σ_j is the standard deviation of such set of values. Regarding the sixth tilt model, which is related to group G_6, its pdf has been set as $\mathrm{p}\left(\theta|G_6\right) = \frac{1}{2\pi}$, since $\theta \in [-\pi, \pi)$.

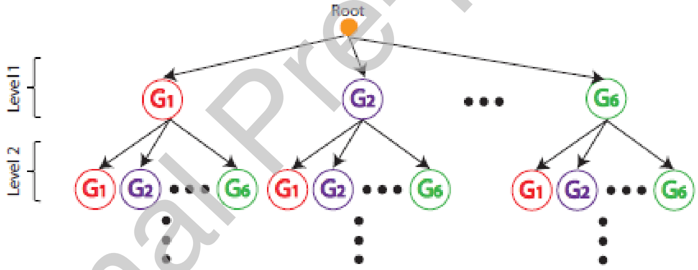

Figure 7: Structure of the PDT.

The most probable classification for all the lines in L' is obtained with the probability decision tree (PDT) illustrated in Figure 7. This PDT has $n_l^{'}$ levels (one per detected line). From each node in each level, six new nodes are obtained in the next level. Thus, the number of nodes in the i-th level is 6^i. In each of the nodes in the i-th level, the probabilities of the i-th line in $L^{'}$ (with tilt value θ_i) of belonging to the considered groups of lines are obtained as:

$$\Pr\left(G_j|\theta_i\right) = \frac{\Pr\left(G_j\right)\mathrm{p}\left(\theta_i|G_j\right)}{\sum_{j=1}^{6}\Pr\left(G_j\right)\mathrm{p}\left(\theta_i|G_j\right)},\ j \in [1,6]\,, \tag{5}$$

where $\Pr\left(G_j\right)$ is the a-priori probability assigned to the j-th group of lines in the node under analysis.

For each node, the prior probabilities are obtained taking into account the following rules:

- There is a maximum of six lines in groups G_1 and G_5.
- There is a maximum of three lines in groups G_2 and G_4.
- There is only one line in group G_3.
- Groups G_1 and G_5 cannot appear simultaneously in the image.
- Groups G_2 and G_4 cannot appear simultaneously in the image.
- Lines belonging to a same group cannot intersect into M_{PF}.

Regarding the first three rules, if the path from the root of the tree to the node under analysis contains the maximum possible number of lines in a group, the PDT is pruned. That is, the hypotheses corresponding to such group of lines are not considered in remaining levels of the tree, which is equivalent to establish null a-priori probabilities for the hypotheses associated to such group of lines from the current node.

Concerning the fourth and fifth rules, when a hypothesis considers that a line belongs to a certain group of lines, the incompatible groups are not considered from that node. That is, the PDT is pruned by establishing null a-priori probabilities in the hypotheses corresponding to incompatible groups in the remaining levels.

Similarly, the last rule allows to prune the tree by analyzing the intersection points between the i-th line and the previous ones, according to how those previous lines have been classified along the path from the root to the node under analysis.

Taking into account these rules, at each node some hypotheses could be discarded. For all the remaining $N_g \leq 6$ hypotheses in such node, the a-priori probability is set as $1/N_g$.

Finally, for each path in the tree, an overall probability is obtained by multiplying the probabilities $\Pr(G_j|\theta_i)$ throughout all the nodes in the path. The path with the highest overall probability is the one used to determine the final classification of the lines.

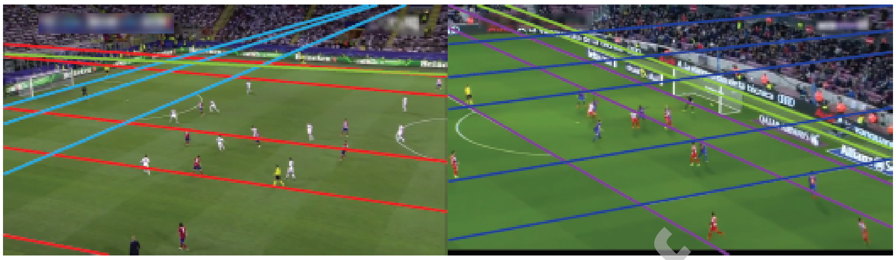

Figure 8: Examples of final line classifications.

Figure 8 shows the final classifications obtained in two images. It can be observed that in both images all the lines have been correctly classified. Moreover, all false detections (one in the left image and two in the right one) have been adequately classified as false lines.

As a result of this classification stage, a new set of n_l straight lines is obtained, $L = \{l_i = (\gamma_i, \theta_i, G_{j,i}, k)\}_{i=1}^{n_l}$, where $G_{j,i}$, with $j \in [1, 5]$, is the group assigned to the i-th line and $k \leq 6$ is an index that indicates the order of the line within the total of lines in its group (from top to bottom in the case of groups G_1 and G_5, from right to left in the case on G_2, and from left to right for lines in group G_4). These indexes are obtained by comparing the intersections of the lines with the image edges. The lines classified in G_6 are not included in this set, so $n_l \leq n_l^{'}$. Additionally, a binary mask of straight lines, M_{SL}, is obtained, whose non-null pixels will be those that voted in the HT to the lines in L. This mask will be used in the validation stage described in section 7.

6. Ellipse modeling

Once the straight lines have been detected and classified, if the halfway line has been detected, the ellipse corresponding to the center circle of the field of play is modeled. First, most of the points of the ellipse appearing in the image are identified (subsection 6.1). Then, an LSF is applied to obtain the parameters of the ellipse and its key-points (subsection 6.2).

In contrast to previous strategies, the proposed one is capable of accurately modeling the ellipse, with the condition that just one of the intersections between the ellipse and the halfway line of the field of play appears in the image (it does not matter that the ellipse does not appear complete in the image). Moreover, the modeling is very computationally efficient.

6.1. Ellipse point detection

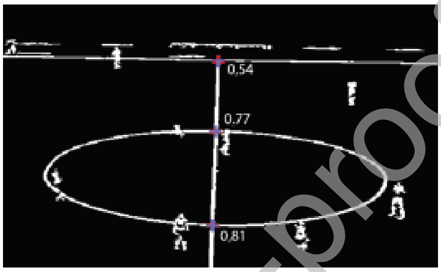

Figure 9: I_{TH} image with the intersection ratios obtained when the structuring element is placed on three candidate intersections. For each candidate, the pixels that have matched the structuring element have been depicted in blue, whereas those not matching it have been depicted in red.

First, a list of intersection points between the line segments in S and the halfway line are computed. If any of the intersections between the ellipse and the halfway line is in the image, it will appear in such list. However, this list can also contain other intersection points (e.g., between the halfway line an a touch line). Thus, to choose the adequate intersection point, for each point in the list, the intersection ratio between I_{TH} and a structuring element with the shape of a cross centered at the coordinates of the point is computed. The point that provides the highest ratio will be chosen as the one that most likely belongs to the ellipse. Figure 9 shows the intersection ratios obtained for the three candidate points in one image. It can be observed that the ratios corresponding to the intersection points on the ellipse are significantly higher than the ratio obtained for the remaining intersection point.

Once the initial point of the ellipse has been established, a region growing algorithm is applied on I_{TH} to identify the remaining ellipse points. To this

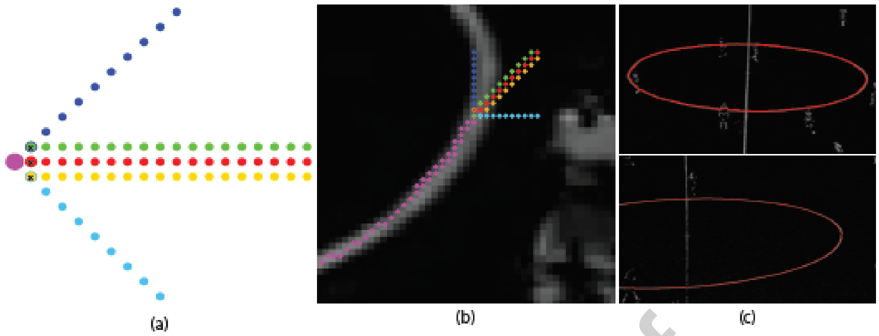

Figure 10: (a) Structuring element, e_{RG}, used to identify the ellipse points: main point in purple; rows of points in blue, green, red, yellow, and cyan; possible next main points marked with an 'x'. (b) Example of the operation of the region growing algorithm (the purple dots denote pixels labeled as ellipse points). (c) Points detected in an image in which the full ellipse appears and in a image in which only a part of the ellipse appears.

end, the structuring element e_{RG}, illustrated in Figure 10.a, has been used, since it has shown to be able to face small discontinuities in the line that defines the ellipse and, in addition, is able to adapt to the curvature of such line. It consists of a main point (purple dot) from which five rows of points extend: three rows in the direction of its main axis (green, red, and yellow dots in the figure) and two rows rotated ±45° from the main axis (blue and cyan dots in the figure). Initially, the element e_{RG} is placed on the pixel corresponding to the intersection point previously identified as part of the ellipse. At each iteration of the algorithm, the averages of the values of I_{TH} corresponding to each of the five rows of points in e_{RG} are computed. In the next iteration, e_{RG} is moved to the position occupied by the first point in the row that has resulted in the highest average (any of the three points marked with an 'x' in the figure), and such new position is added to the set of ellipse points. Additionally, if the highest average is obtained from the blue or green rows, e_{RG} is not only displaced but also rotated $-45°$. Similarly, if the highest average results from the yellow or cyan rows, e_{RG} is rotated 45°. In this way, the element is able of adapting to the curvature of the ellipse at each point. Figure 10.b illustrates an example of how the points of the ellipse are correctly identified and how e_{RG} has rotated to adapt to the curvature of the ellipse.

The region growing process finishes when the new point to be added to the set of ellipse points is already in such set, or when the image limits are reached. In this second case, to try to obtain more ellipse points, a new region growing process is applied from the initial point, but in opposite direction. Figure 10.c shows the set of points obtained in an image in which the full ellipse appears, whereas Figure 10.d illustrates the result in an image in which only a part of the ellipse can be seen. Although in both examples there are players located on the ellipse line, the results are successful.

As a result of this process, a new binary mask, M_{CC}, indicating the positions of the pixels belonging to the ellipse is obtained.

6.2. Estimation of key-points

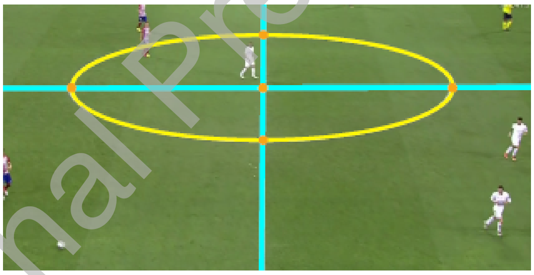

Figure 11: Estimation of key-points in the center circle: Estimated ellipse in yellow, central lines (longitudinal and transverse) in cyan; and ellipse key-points in orange.

Applying an LSF algorithm to the set of ellipse points in M_{CC}, the general quadratic curve of the ellipse is estimated. Then, the following five parameters describing the ellipse [41] are obtained: the center, $C = (x_0, y_0)$; the semi-major, a, and semi-minor, b, axes; and the counterclockwise angle of rotation from the x-axis to the major axis of the ellipse, α.

The set of key-points in the center circle, P_{CC}, which is composed by the points P_{18}, P_{29}, P_{30}, P_{31}, and P_{42} (see appendix) is easily obtained by computing the intersections between the ellipse equation and the two central lines in the field of play (the halfway line and the the longitudinal central line). Figure 11 illustrates an example of the results obtained.

6.3. Validation

To determine whether the ellipse modeling has been successful, the three-step validation process described in section 7 is applied from the points in P_{CC}. If the modeling is not successful, it is carried out again, using as a starting point the next intersection point with the highest ratio (from the list obtained at the beginning of the process). If none of the intersection points result in a successful modeling, P_{CC} and M_{CC} will be empty.

7. Hypothesis validation

The lines in L are classified into the five groups described in section 5 and, moreover, they are sorted (from top to bottom in the case of groups G_1 and G_5, from right to left in the case of G_2, and from left to right for lines in group G_4). However, given a group, it is not known which line of such group corresponds to each of the detected ones. So, multiple hypotheses must be considered.

Let $P_T = \{P_L, P_{CC}\}$ be a set of image points, where P_L is the set of intersection points between the longitudinal (groups G_1, G_3, and G_5) and transverse (groups G_2 and G_4) lines in L. Each hypothesis will consist on establishing matches between the points in P_T and key-points in the model of the field of play, I_{FM} (see appendix). If there is a minimum of four points in P_T, it will be possible to estimate the homography matrix, $\mathbf{H}$, that relates the points in I_{FM} to the points in I as $p_I = \mathbf{H}\ p_{FM}$ [42], where p_I is a point in I, and p_{FM} is the corresponding point in I_{FM}.

A validation process is carried out to identify the correct hypothesis, which consists of the following three steps: camera position analysis, field of play compatibility, and line mark compatibility. Only the correct hypothesis should satisfy the conditions in all three steps, avoiding the delivery of wrong results.

In some cases, lines defined by the billboards or by any of the three circles in the field of play (see appendix) can be erroneously classified as straight line marks. In these cases, the images under analysis will not pass the validation process. However, generally, these erroneously classified lines will be the least important ones. Therefore, since the lines are sorted by the reliability of their detections (as stated in section 5.1), the least reliable line will be discarded and the validation process will be applied again. If the validation process has not yet been completed, more lines will be discarded, one at a time, until a valid result is reached (provided that the minimum number of points in P_T is enough to estimate the homography).

7.1. Camera position analysis

Let $\mathbf{H}$ be the homography matrix under analysis and let x_{IP} be the vertical coordinate of the intersection point of G_{31} and the bisector of the cone (from now, vision-cone) resulting from applying the transform $\mathbf{H}^{-1}$ to I. Since the Master Camera is placed approximately on the extension of g_{31} and its roll is supposed to be null, x_{IP} must approximately match the horizontal coordinate of the Master Camera, x_{MC}, which is already known. However, to deal with possible inaccuracies in the estimation of $\mathbf{H}$, a hypothesis is considered valid if x_{IP} matches a Normal distribution defined as $N(x_{MC}, \sigma_{MC})$. That is [43]:

$$|x_{IP} - x_{MC}| \leq 3 \cdot \sigma_{MC} \tag{6}$$

In the experiments performed, the key-point P_{56} (see appendix) has been set as the origin of the coordinates and it has been verified that the results are successful if the standard deviation of the distribution is set as $\sigma_{MC} = 0.1 \cdot x_{MC}$.

Figure 12 shows some results obtained in the case of an image with three lines in G_2 and three lines in G_5. Only the second hypothesis satisfies equation 6. However, as it is not correct, it will be discarded in the remaining validation steps.

7.2. Field of play compatibility

In the second validation step, M_{PF} and a new binary mask, Θ, resulting from applying the homography $\mathbf{H}$ to I_{FM}, are compared for each hypothesis.

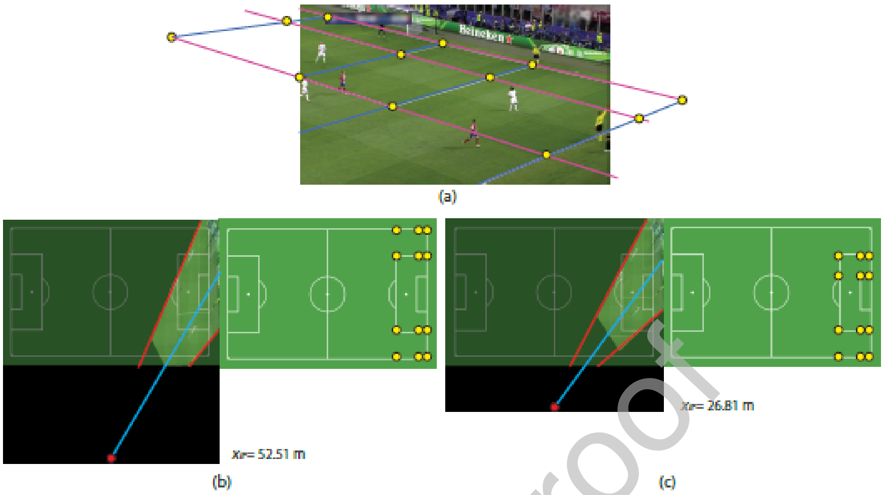

Figure 12: (a) I with the detected lines in purple and blue, and the intersection points, P_I, in yellow. (b-c) Two of the fifteen possible hypotheses from the points in P_I. At the left side of each hypothesis: I transformed to I_{FM}, where the red lines delimit the vision-cone, the cyan line is the bisector of the vision-cone, and the red dot is the intersection point between said bisector and the halfway line. At the right side of each hypothesis: I_{FM} and the points considered in the hypothesis. According to the x_{IP} values obtained for each hypothesis, only the second one will be considered for the remaining validation steps (in this experiment $x_{MC} = 28$ m).

A hypothesis will satisfy this step if

$$F_f = \frac{2 \cdot tp_f}{2 \cdot tp_f + fp_f + fn_f} > Th_f, \tag{7}$$

where F_f is the harmonic mean, usually called F-score, of the recall ($r_f = \frac{tp_f}{tp_f + fn_f}$) and the precision ($p_f = \frac{tp_f}{tp_f + fp_f}$) values obtained from the comparison, $tp_f = \|M_{PF} \circ \Theta\|_0$, $fn_f = \left\|M_{PF} \circ \bar{\Theta}\right\|_0$, and $fp_f = \left\|\bar{M}_{PF} \circ \Theta\right\|_0$ ($\circ$ being the Hadamard product, $\|\cdot\|_0$ the zero norm defined as the number of nonzero elements in a matrix, and $\bar{\cdot}$ the complementary of a binary mask).

The experiments performed have shown that the correct hypotheses always satisfy that $F_f > 0.9$. Consequently, a value of $Th_f = 0.9$ was finally chosen.

Figure 13 shows some results obtained with this second validation step in the case of an image with six straight lines (three in G_2, two in G_5, and g_{31}),

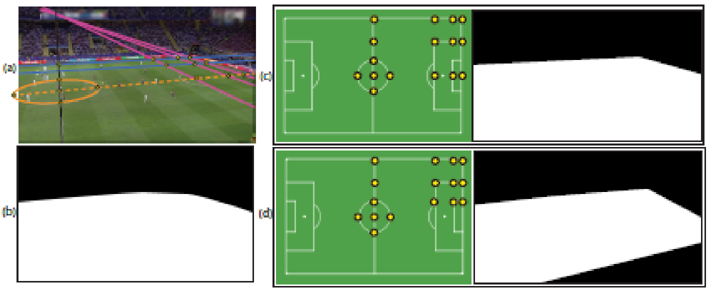

Figure 13: (a) I with the detected lines, the estimated ellipse, the longitudinal central line, and the intersection points in P_I. (b) M_{PF}. (c-d) Two of the fifteen possible hypotheses from the points in P_I. At the left side of each hypothesis: I_{FM} and the points considered in the hypothesis. At the right side of each hypothesis: Θ.

and the center ellipse. The first hypothesis analyzed in this figure is the correct one and has satisfied equation 7. On the other hand, the second hypothesis has not satisfied the equation.

7.3. Line mark compatibility

In the last validation step, the binary mask Ψ, resulting from applying the homography $\mathbf{H}^{-1}$ to $M_{SL} \circ M_{CC}$, and binary line mark mask, Λ, corresponding to the lines in I_{FM} that are within the vision-cone, are compared. A hypothesis will satisfy this step if

$$F_l = \frac{2 \cdot tp_l}{2 \cdot tp_l + fp_l + fn_l} > Th_l, \tag{8}$$

where F_l is another harmonic mean computed from: $tp_l = \left\|\Lambda \circ \Psi\right\|_0$, $fn_l = \left\|\Lambda \circ \bar{\Psi}\right\|_0$, and $fp_l = \left\|\bar{\Lambda} \circ \Psi\right\|_0$. Because of the inaccuracies in the estimation of $\mathbf{H}$, the values of F_l that are obtained for the correct hypotheses are not as high as those obtained for F_f. Consequently, a threshold $Th_l = 0.65$ has been chosen, since it has proven to be adequate in all the performed experiments.

Figure 14 illustrates a couple of hypotheses corresponding to the same image analyzed in Figure 12. The first hypothesis (Figure 14.b) is the correct one and has resulted in $F_l = 0.95$. The second hypothesis (Figure 14.c) is not correct and has resulted in $F_l = 0.60$.

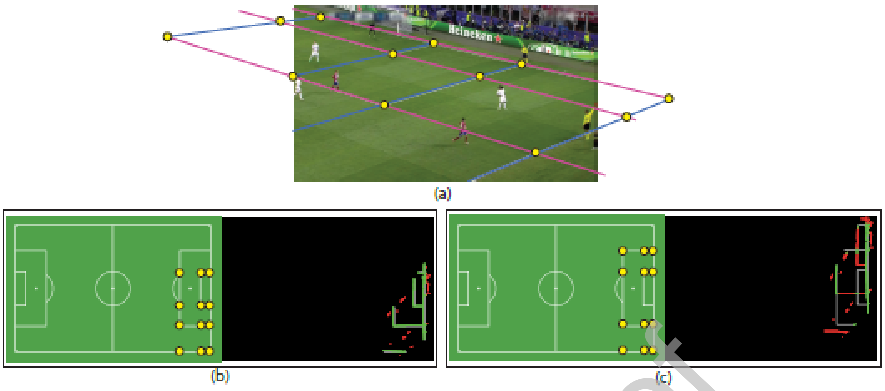

Figure 14: (a) I with the detected lines in purple and blue, and the intersection points, P_I, in yellow. (b-c) Two of the fifteen possible hypotheses from the points in P_I. At the left side of each hypothesis, I_{FM} and the points considered in the hypothesis. At the right side of each hypothesis, the comparison between Ψ and Λ: true positives (tp_l) in green, false positives (fp_l) in red, and false negatives (fn_l) in white.

8. Results

The proposed strategy has been tested on 1800 frames belonging to twelve sequences recorded in different stadiums and corresponding to different European championship finals. These stadiums have been chosen mainly because of 460 the different locations of the Master Camera in each of them (different heights and different x_{MC} values), but also because of their different grass colors. Nevertheless, note that the images used to construct the tilt models (described in section 5) do not belong to any one of these three stadiums, but to many more stadiums covering a wide range of positions of the Master Camera.

465 Table 1 summarizes the main characteristics of the test sequences.

Table 1: Main characteristics of the test sequences.

Name	Resolution	X_{MC} (meters)
Stadium 1	1280 × 720	28
Stadium 2	1280 × 720	52
Stadium 3	1280 × 720	19

For each stadium, four types of sequences have been considered, focused on a different corner of the field of play: Lower-left area (type 1), upper-left area (type 2), upper-right area (type 3), and lower-right area (type 4). Additionally, to analyze the performance of the proposed strategy from key-points belonging to the center circle, some of these sequences include images showing the center circle: type 2 in stadiums 1 and 3, and type 3 in stadium 2.

All the test sequences, the model of the field of play, and the images used for constructing the tilt models are available online[2], as well as the final (Stage 3) and intermediate (stages 1 and 2) results.

8.1. Computational analysis

The computational analysis has been carried out on a 4 GHz CPU with 16 GB RAM. Furthermore, to achieve a faster processing speed, the proposed strategy has been implemented on a general-purpose graphics processing unit (GPGPU) nVidia GeForce GTX 1080.

Table 2: Computational analysis in each stage of the proposed system.

Stage	Percentage of cost	Mean processing time per image
Field of play segmentation	14.44%	5.52 ms
Line segment detection	39.58%	15.13 ms
Stage 1	13.37%	5.11 ms
Stage 2	19.80%	7.57 ms
Stage 3	12.82%	4.90 ms
Overall	100%	38.23 ms

The summary of the results obtained is shown in Table 2. These results show that the line segment detection performed in the preprocesing stage is responsible for the bulk of the computational cost. This is because the the

[2] www.gti.ssr.upm.es/data

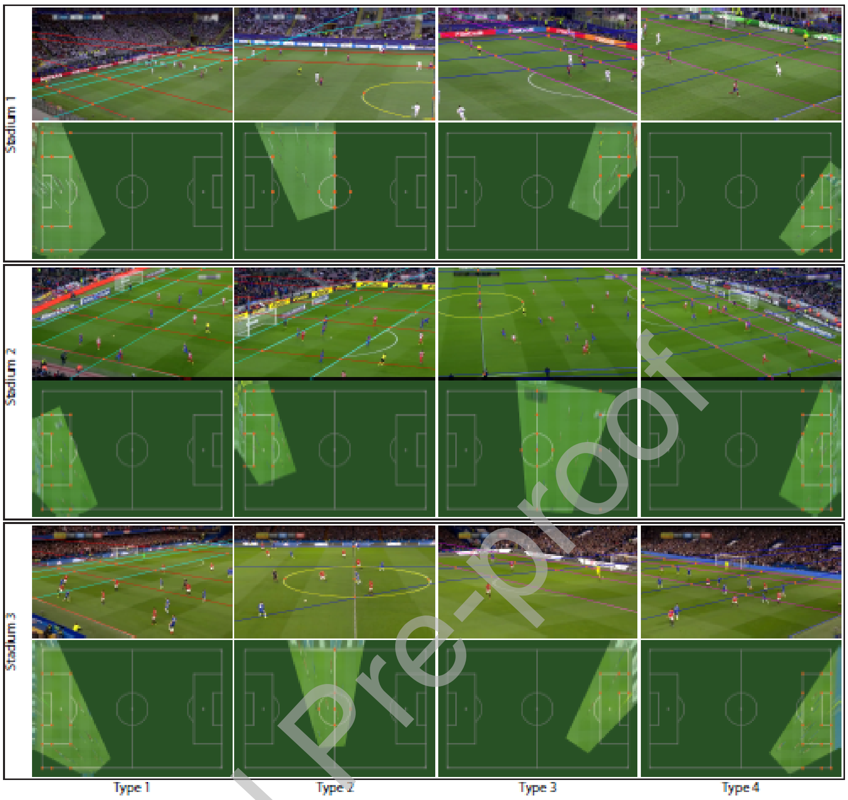

Figure 15: Some representative results. Upper images: I, lines in L, and points used to estimate $\mathbf{H}$. Bottom images: projection of I to I_{FM} and points used to estimate $\mathbf{H}$.

ELLSF involves iterative operations that cannot be easily parallelized. The operations required for detecting the points of the center circle in Stage 2 present the same problem. On the contrary, although stages 1 and 3 are the ones that require the most operations per image, they are highly parallelizable in the GPGPU. Consequently, as the table shows, these two stages involve only 25% of the total system cost. Regarding the segmentation of the field of play in the preprocesing stage, the computation of the convex hull it includes is not amenable to easily parallelization and, hence, is carried out in the CPU.

The mean overall cost per image is 38 ms, which makes the proposed system suitable for being used in real-time applications.

8.2. Quality analysis

Figure 15 illustrates some representative results in each type of view of each test sequence. Although all these results are successful, in very few cases (e.g., type 2 in stadium 3) false line marks have been obtained. In these cases, the intersection points corresponding to the false lines have been discarded. Consequently, they have not been considered in the estimation of **H**.

Table 3: Summary of results.

Name	Type	Num. of frames	Failures		Without ellipse	With ellipse
			Num.	Perc.		
Stadium 1	1	150	0	0%	150	0
	2	150	46	31%	54	50
	3	150	0	0%	150	0
	4	150	0	0%	150	0
Stadium 2	1	150	0	0%	150	0
	2	150	0	0%	150	0
	3	150	21	14%	62	67
	4	150	0	0%	150	0
Stadium 3	1	150	0	0%	150	0
	2	150	35	23%	110	5
	3	150	0	0%	150	0
	4	150	5	3%	145	0
Total		1800	107	6%	1571	122

Most of the results obtained have been successful. However, in some cases it has not been possible to determine **H**. Table 3 summarizes, for each test sequence, the number of analyzed images, the amount and percentage of failures, the number of results from only straight lines, and the amount of results from straight lines and the ellipse. Additionally, Figure 16 illustrates the results obtained for each image in the test sequences. These results show that the registration has been completely successful in many of the sequences (8 of 12),

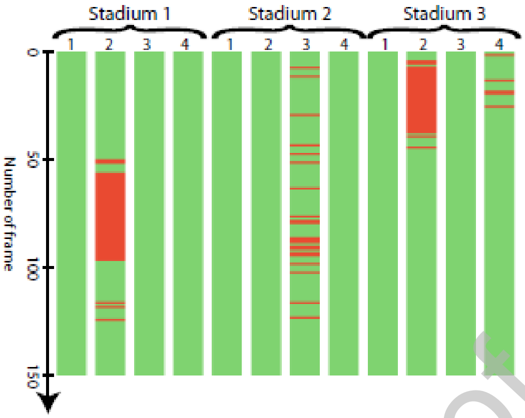

Figure 16: Results obtained for each image in the test sequences. Green and red denote, respectively, successful and unsuccessful results.

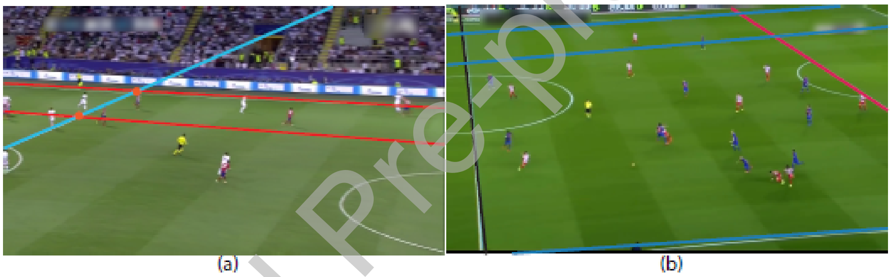

Figure 17: (a) Image without enough key-points for performing the registration. (b) Image with inaccurate line detections due to radial distortion of the camera.

whereas it has been unsuccessful only in 6% of the analyzed images. Most of these unsuccessful results correspond to images without enough visible line marks (see Figure 17.a). These images appear when the camera is moving from one area to the center of the field of play (type 2 in stadiums 1 and 3, and type 3 in stadium 2). In the third sequence (type 3) in stadium 2 the camera also moves from the right area to the center of the field of play. However, in this sequence the amount of unsuccessful results is lower because the camera is significantly farther from the field of play than in the other stadiums ($x_{MC} = 52$). Therefore, the amount of visible line marks is enough in most images. Nevertheless, because the camera is so far away, there are more precision errors (mainly due to the radial distortion of the camera) than in other sequences. This is the cause of the failures (21 images) obtained in this sequence (see Figure 17.b, where the inaccuracy in the estimation of both the halfway line, g_{31}, and the upper touch line, g_{51}, can be observed). Finally, it must be mentioned that the remaining errors (type 2 in stadium 1 and type 4 in stadium 3) are isolated failures that are mainly due to inaccurate estimates of straight lines due to the location of the players on the field of play.

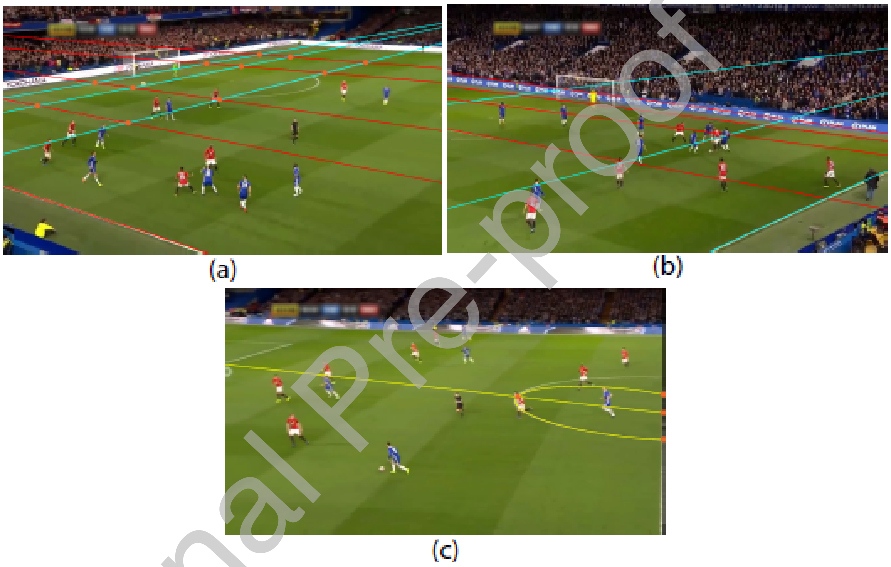

Figure 18: (a) Image in which the upper touch line has not been detected because it cannot be distinguished from the grass. (b) Wrong classification (groups G_1 and G_4 have been confused with groups G_2 and G_5). (c) Unsuccessful modeling of the ellipse.

Regarding the results obtained from the intermediate stages, the following conclusions have been obtained. Line detection and classification (Stage 1) have been successful in most images. One typical source of errors has been the non-detection of lines that are either too short or very similar to grass (see Figure 18.a). However, these errors generally do not prevent the determination of **H**, since there are usually enough lines detected in all images. Another typical source of errors have been the incorrect classifications when the Master Camera is very rotated (see Figure 18.b), since in these cases groups G_1 and G_4

may be confused with groups G_2 and G_5, respectively. To solve this problem, if the estimation of $\mathbf{H}$ has not been possible, Stage 3 is applied again after swapping the lines in these two pairs of groups. Regarding Stage 2, as indicated in section 6, ellipse modeling is performed only on images that show any of the intersections between the center circle and the halfway line. This occurs in 53 images of stadium 1 (type 2 sequence), 84 images of stadium 2 (type 3 sequence), and 12 images of stadium 3 (type 2 sequence). In stadiums 1 and 2 there are no errors. However, in stadium 3 there are two errors due to players standing on the line that defines the center circle (see Figure 18.c). These players have prevented a significant amount of ellipse points from being detected. Consequently, ellipse modeling has failed.

9. Conclusions

A robust strategy for the automatic registration of soccer images on a model of the field of play has been proposed, which is suitable for being used in real-time applications.

The straight line marks are detected and classified according to their tilt. The detection is performed from a set of line segments that does not contain noisy edges and that are completely centered in the line marks. The classification is carried out by means of a probabilistic decision tree that allows identifying the most probable classification for the set of all detected lines. To provide results not only in images of the penalty areas but also in images showing the center circle, the ellipse corresponding to said center circle is also modeled. This modeling is done from just one of the intersections between the ellipse and the halfway line, even if the center circle does not appear complete in the image. Finally, to establish the set of key-points determining the homography matrix that relates the image and the model, a three-step validation stage is applied, which allows determining when the registration is not possible.

The quality of the strategy has been assessed on twelve sequences corresponding to three stadiums with different characteristics. The obtained results

560 have proven that in most of the analyzed images (94%) the registration has been successful.

Appendix A. Field of play model

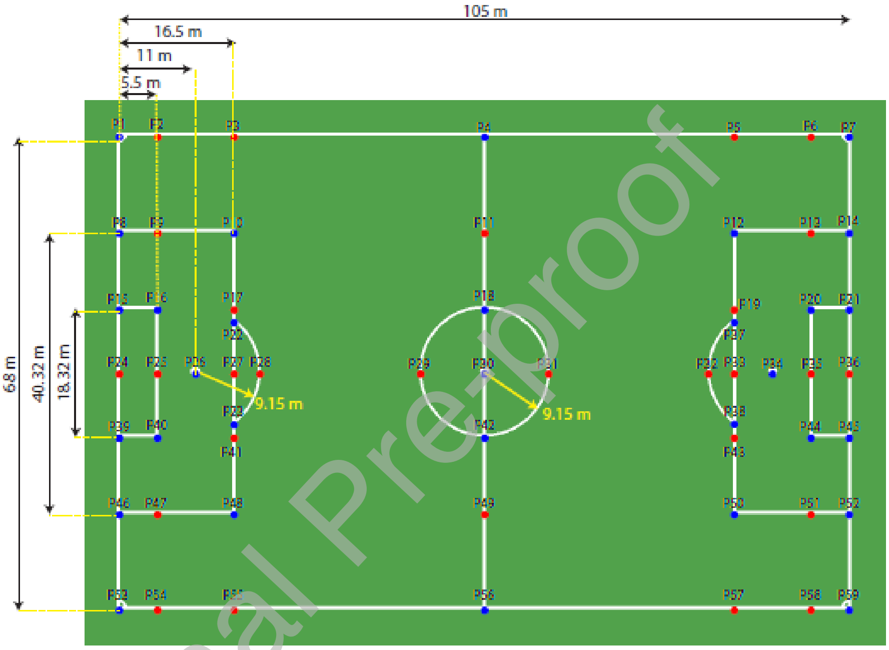

Figure A.19: Model of the field of play, I_{FM}, with its dimensions and 59 key-points. Blue dots denote key-points obtained from the intersections between the line marks. Red dots denote key-points obtained from intersections between line marks and extensions of other line marks.

Figure A.19 illustrates the model of the field of play, I_{FM}, that has been used in the proposed registration strategy. The dimensions of this model have 565 been established according to the rules of The International Football Association Board (IFAB) [40], particularizing in the case of stadiums of categories 2 and 3 considered by the Union of European Football Associations (UEFA) in the Champions League [44, 45]. As the figure shows, the line marks in I_{FM} allow to obtain a set of 59 key-points, $P_{FM} = \{P_i\}_{i=1}^{59}$, that can be used for the 570 registration of real images.

Acknowledgements

This work has been partialy supported by the Ministerio de Economa, Industria y Competitividad (AEI/FEDER) of the Spanish Government under project TEC2016-75981 (IVME).

References

[1] A. E. Hassanien, Machine learning-based soccer video summarization system., in: International Conference in Multimedia, Computer Graphics and Broadcasting, 2018.

[2] R. da Silva, S. R. Dahmen, Universality in the distance between two teams in a football tournament, Physica A: Statistical Mechanics and its Applications 398 (2014) 56–64.

[3] M. N. Ali, M. Abdullah-Al-Wadud, S.-L. Lee, An efficient algorithm for detection of soccer ball and players, in: Conference on Signal and Image Processing (SIP), 2012, pp. 1–8.

[4] A. Bialkowski, P. Lucey, P. Carr, Y. Yue, S. Sridharan, I. Matthews, Large-scale analysis of soccer matches using spatiotemporal tracking data, in: IEEE International Conference on Data Mining, IEEE, 2014, pp. 725–730.

[5] R. M. Barros, M. S. Misuta, R. P. Menezes, P. J. Figueroa, F. A. Moura, S. A. Cunha, R. Anido, N. J. Leite, Analysis of the distances covered by first division Brazilian soccer players obtained with an automatic tracking method, Journal of sports science & medicine 6 (2) (2007) 233.

[6] M.-S. Hosseini, A.-M. Eftekhari-Moghadam, Fuzzy rule-based reasoning approach for event detection and annotation of broadcast soccer video, Applied Soft Computing 13 (2) (2013) 846–866.

[7] Q. Yao, A. Kubota, K. Kawakita, K. Nonaka, H. Sankoh, S. Naito, Fast camera self-calibration for synthesizing free viewpoint soccer video, in:

IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2017, pp. 1612–1616.

[8] X. Tong, J. Liu, T. Wang, Y. Zhang, Automatic player labeling, tracking and field registration and trajectory mapping in broadcast soccer video, ACM Transactions on Intelligent Systems and Technology 2 (2) (2011) 15.

[9] A. Bozorgpour, M. Fotouhi, S. Kasaei, Robust homography optimization in soccer scenes, in: Iranian Conference on Electrical Engineering, IEEE, 2015, pp. 787–792.

[10] N. Homayounfar, S. Fidler, R. Urtasun, Soccer field localization from a single image, arXiv preprint arXiv:1604.02715.

[11] J.-B. Hayet, J. H. Piater, J. G. Verly, Fast 2d model-to-image registration using vanishing points for sports video analysis, in: IEEE International Conference on Image Processing, Vol. 3, IEEE, 2005, pp. III–417.

[12] N. Homayounfar, S. Fidler, R. Urtasun, Sports field localization via deep structured models, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5212–5220.

[13] A. Linnemann, S. Gerke, S. Kriener, P. Ndjiki-Nya, Temporally consistent soccer field registration, in: IEEE International Conference on Image Processing (ICIP), IEEE, 2013, pp. 1316–1320.

[14] Q. Yao, K. Nonaka, H. Sankoh, S. Naito, Robust moving camera calibration for synthesizing free viewpoint soccer video, in: IEEE International Conference on Image Processing, IEEE, 2016, pp. 1185–1189.

[15] L. Alvarez, V. Caselles, Homography estimation using one ellipse correspondence and minimal additional information, in: IEEE International Conference on Image Processing, IEEE, 2014, pp. 4842–4846.

[16] P. Mukhopadhyay, B. B. Chaudhuri, A survey of hough transform, Pattern Recognition 48 (3) (2015) 993–1010.

[17] L. Xu, E. Oja, P. Kultanen, A new curve detection method: randomized hough transform (rht), Pattern recognition letters 11 (5) (1990) 331–338.

[18] J. R. Bergen, H. Shvaytser, A probabilistic algorithm for computing hough transforms, Journal of algorithms 12 (4) (1991) 639–656.

[19] Y. Xie, Q. Ji, A new efficient ellipse detection method, in: IEEE International Conference on Pattern Recognition, Vol. 2, IEEE, 2002, pp. 957–960.

[20] X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, K. W. Wan, Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video, in: ACM international conference on Multimedia, ACM, 2003, pp. 11–20.

[21] X. Yu, H. W. Leong, C. Xu, Q. Tian, A robust hough-based algorithm for partial ellipse detection in broadcast soccer video, in: IEEE International Conference on Multimedia and Expo, Vol. 3, IEEE, 2004, pp. 1555–1558.

[22] K. Wan, L. Joo-Hwee, C. Xu, X. Yu, Real-time camera field-view tracking in soccer video, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 3, IEEE, 2003, pp. III–185.

[23] Y. Wu, H. Wang, F. Tang, Z. Wang, Efficient conic fitting with an analytical polar-n-direction geometric distance, Pattern Recognition 90 (2019) 415–423.

[24] F. Wang, L. Sun, B. Yang, S. Yang, Fast arc detection algorithm for play field registration in soccer video mining, in: IEEE International Conference on Systems, Man and Cybernetics, Vol. 6, IEEE, 2006, pp. 4932–4936.

[25] T. Battikh, I. Jabri, Camera calibration using court models for real-time augmenting soccer scenes, Multimedia Tools and Applications 51 (3) (2011) 997–1011.

[26] M. Alemán-Flores, L. Álvarez, L. Gómez, P. Henríquez, L. Mazorra, Camera calibration in sport event scenarios, Pattern Recognition 47 (1) (2014) 89–95.

[27] C. B. Duane, Close-range camera calibration, Photogramm. Eng 37 (8) (1971) 855–866.

[28] M. Hoernig, M. Herrmann, B. Radig, Real time soccer field analysis from monocular tv video data, in: International Conference on Pattern Recognition and Image Analysis, Vol. 2, The Russian Academy of Sciences, 2013, pp. 567–570.

[29] A. Ekin, A. M. Tekalp, R. Mehrotra, Automatic soccer video analysis and summarization, IEEE Transactions on Image processing 12 (7) (2003) 796–807.

[30] V. Ngo, J. Cai, Converting 2D soccer video to 3D cartoon, in: IEEE International Conference on Control, Automation, Robotics and Vision, IEEE, 2008, pp. 103–107.

[31] M. Fotouhi, A. Bozorgpour, S. Kasaei, Automatic soccer field line recognition by minimum information, in: IEEE International Symposium on Artificial Intelligence and Signal Processing (AISP), IEEE, 2015, pp. 136–142.

[32] M. Manafifard, H. Ebadi, H. A. Moghaddam, Multi-player detection in soccer broadcast videos using a blob-guided particle swarm optimization method, Multimedia Tools and Applications 76 (10) (2017) 12251–12280.

[33] U. Rao, U. C. Pati, A novel algorithm for detection of soccer ball and player, in: IEEE International Conference on Communications and Signal Processing (ICCSP), IEEE, 2015, pp. 0344–0348.

[34] Y. Gong, L. T. Sin, C. H. Chuan, H. Zhang, M. Sakauchi, Automatic parsing of TV soccer programs, in: IEEE International Conference on Multimedia Computing and Systems, IEEE, 1995, pp. 167–174.

[35] J. Bu, S. Lao, L. Bai, Automatic line mark recognition and its application in camera calibration in soccer video, in: IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2011, pp. 1–6.

[36] V. A. Ngo, W. Yang, J. Cai, Accurate playfield detection using area-of-coverage, in: IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, 2010, pp. 3441–3444.

[37] D. Quilón, R. Mohedano, C. Cuevas, N. García, Unsupervised high-quality soccer field segmentation, in: IEEE International Symposium on Consumer Electronics (ISCE), IEEE, 2015, pp. 1–2.

[38] P. Kovesi, Edge linking and line segment fitting.
URL `http://www.peterkovesi.com/matlabfns/index.html`

[39] E. Kauder, Genesis of the marginal utility theory: from Aristotle to the end of the eighteenth century, The Economic Journal 63 (251) (1953) 638–650.

[40] Laws of the Game 2018/19, The International Football Association Board, IFAB, 2018.

[41] H. Dong, D. K. Prasad, I.-M. Chen, Accurate detection of ellipses with false detection control at video rates using a gradient analysis, Pattern Recognition 81 (2018) 112–130.

[42] W.-H. Song, H.-G. Jung, I.-Y. Gwak, S.-W. Lee, Oblique aerial image matching based on iterative simulation and homography evaluation, Pattern Recognition 87 (2019) 317–331.

[43] G. Gallego, C. Cuevas, R. Mohedano, N. García, On the Mahalanobis distance classification criterion for multidimensional normal distributions, IEEE Transactions on Signal Processing 61 (17) (2013) 4387–4396.

[44] UEFA Stadium Infraestructure Regulations, Union of European Football Associations, UEFA, 2018.

[45] Regulations of the UEFA Champions League 2018-21 Cycle, Union of European Football Associations, UEFA, 2018.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Carlos Cuevas received the Ingeniero de Telecomunicación degree and the Doctor Ingeniero de Telecomunicación degree (Doctoral Graduation Award) from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2006 and 2011, respectively. Since 2006, he has been a member of the Image Processing Group of the UPM.

Daniel Quilón received the Ingeniero de Telecomunicación degree from the Universidad Politécnica de Madrid, Madrid, Spain, in 2018. He has been collaborating with the (Image Processing Group in the UPM since 2015. Since 2013 he has been working in the audiovisual sector, building real time graphics for TV.

Narciso García received the Ingeniero de Telecomunicación degree and the Doctor Ingeniero de Telecomunicación degree (Doctoral Graduation Award) from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 1976 and 1983, respectively. Since 1977, he has been a member of the faculty of the UPM. He leads the Image Processing Group of the UPM.