Noname manuscript No.
(will be inserted by the editor)

A Vision-Based System to Support Tactical and Physical Analyses in Futsal

Pedro H. C. de Pádua · Flávio L. C. Pádua · Marconi de A. Pereira · Marco T. D. Sousa · Matheus B. de Oliveira · Elizabeth F. Wanner

Received: date / Accepted: date

Abstract This paper presents a vision-based system to support tactical and physical analyses of futsal teams. Most part of the current analyses in this sport are manually performed, while the existing solutions based on automatic approaches are frequently composed by costly and complex tools, developed for other kind of team sports, making it difficult their adoption by futsal teams. Our system, on the other hand, represents a simple yet efficient dedicated solution, which is based on the analyses of image sequences captured by a single stationary camera used to obtain top-view images of the entire court. We use adaptive background subtraction and blob analysis to detect players, as well as particle filters to track them in every video frame. The system determines the distance traveled by each player, his/her mean and maximum speeds, as well as generates heat maps that describe players' occupancy during the match. To present the collected data, our system uses a specially developed mobile application. Experimental results with image sequences of an official match and a training match show that our system provides data with global mean tracking errors below 40 cm, demanding on 25 ms to process each frame and, thus, demonstrating its high application potential.

Pedro H. C. de Pádua
DECOM, CEFET-MG, Av. Amazonas, 7675, 30510-000, MG, Brazil
E-mail: pedhenrique@decom.cefetmg.br

Flávio L. C. Pádua
E-mail: cardeal@decom.cefetmg.br

Marconi de A. Pereira
E-mail: marconi@ufsj.edu.br

Marco T. D. Sousa
E-mail: mtdsousa@decom.cefetmg.br

Matheus Barcelos de Oliveira
E-mail: mboliveira@decom.cefetmg.br

Elizabeth F. Wanner
efwanner@decom.cefetmg.br

Keywords Tactical Analysis · Physical Analysis · Futsal · Computer Vision · Player Tracking · Mobile Applications.

1 Introduction

The professional sport activity has become, on the last decades, more and more competitive. In such a high performance level, small modifications on actions of athletes can produce better results and lead to victory [24,30]. To achieve this goal, team staffs play an important role by analyzing athletes performances. Those professionals constantly study the decisions of athletes to verify possible improvements that can benefit them both tactically and physically [37,44].

In futsal, such analyses are fundamental to understand what is happening in the game and to identify errors. Through those observations, coaches can perceive tactical patterns used by teams, refine strategies, verify players physical efficiency and better adapt training routines [49,33,30].

To correctly identify the tactical patterns and to verify player's physical aspects, it is necessary, at first, to correctly estimate the positions of the athletes at a given instant of time and, consequently, track them [30]. Through this estimate, one can define the trajectories of players and extract key statistics for the analyses.

Most part of the current analyses, however, are manually performed, conducted by staff members or specialized companies [16,30]. The matches are recorded in video and reviewed exhaustively so that observations are made, registered and passed later to coaches. Those approaches are prone to human error, demanding on significant time and financial costs. On the other hand, some technological solutions based on automatic approaches have been developed, capable of speeding up the statistics extraction process and helping teams on match analyses [5,3,1,4]. Unfortunately, they are, in most cases, composed by costly and

complex tools, developed for other kinds of sports (e.g. soccer and basketball), making it difficult their adoption by futsal teams. Additionally, some of the information obtained is only available several days after the game, what limits their use by technical staff on post-match training [16].

In this context, this work proposes a vision-based system to support tactical and physical analyses of futsal teams, by automatically detecting and tracking players in video frames with low human intervention. A vision-based solution to the domain of futsal is characterized by several challenges. These include, but are not limited to, illumination variation, camera lens distortion, appropriate coverage of the scene of the match, usage of bright reflective materials on the court, players' occlusions, players with very fast motion dynamics, shadows cast by the futsal gym and players, object shape deformation due to fast motion, objects we consider as 'noise' such as the fans, the coaches standing on the sideline, the reserve players warming up, ground staff and so on.

The tactical data provided by the proposed system consist of player's court occupancy heat maps, while the physical data consist of mean and maximum speeds, as well as the distance covered by each one of the athletes. In order to detect players in images, we use an adaptive background subtraction method based on mixture of Gaussians [34,50] and blob analysis to check geometric constraints of the blobs in the resulted binary images. To track multiple players in successive frames and estimate their positions at a given instant of time, we use particle filters [34]. Each player detected is automatically tracked by a separate particle filter. For the sake of solving data association between detections and trackers in each frame, our system uses the well-known Hungarian Algorithm [34,25]. If a detection is associated to a tracker, it is used to guide the particles of the associated tracker. Otherwise, filter's prediction and the appearance model of the player, based on color histograms, are used to estimate the position of the player at that time. The initial identification of the players is made in a semi-automatic way: an operator of the system is responsible to provide the identity of each player's tracker. With the position data of players extracted and their identification, we can extract tactical and physical data of the athletes. Finally, the data are presented to coaches and trainers through a mobile application specially developed to run on smartphones and tablets, that queries the database of the system.

That said, the main contribution of this work consists in to present and validate a complete, simple and effective vision-based system to support tactical and physical analyses in futsal. Unlike most part of the previous works, which are based on complex architectures constituted of multiple cameras (eventually combined with other types of sensors) and developed for other kinds of sports, our solution utilizes a single stationary camera that monitors the entire court area and captures top-view images, consequently reducing the undesired effects of occlusions among players. The lack of studies based on simpler image acquisition systems turns our work an important contribution on its research area. Moreover, our system also presents a semi-automatic approach to segment actions of interest (e.g. free kicks, goals, passes, fouls, among others) during the game, what differentiates it from most previous solutions that usually provide this kind of information only *a posteriori*, limiting their applications by technical staff on post-match analysis. Instead, by using our solution, one member of the technical staff runs a command in a mobile application at the moment when an important game action occurs. The system can then edit a video file that contains some anterior and posterior actions along with the one of interest and make it available for streaming in the mobile device. The coaches and trainers can show this video slice to players to correct team flaws or to show weaknesses of the opponent team. Finally, we believe that our paper contributes to demonstrate the actual applicability of some well-known computer vision methods, when they are combined to estimate critical metrics used to analyze tactical and physical performances of a futsal team. As far as we know, this is the first work in the literature to propose such a simple and dedicated system for futsal, which might be of special interest to elite coaches and sports science researchers. It is important to emphasize that futsal is a very particular and dynamic team sport, whose players are often in touch, concentrated in small areas and moving quickly. Therefore, a number of difficulties have to be faced and overcome in order to develop a computational solution capable of providing useful data for the analyses. This means that vision-based solutions developed for other kinds of sports are not simply adapted to futsal. Certainly, from this kind of study, new research efforts could be derived in order to develop more robust computer vision algorithms to, for example, people detection and tracking in very complex contact scenarios, such as the ones observed in futsal matches.

The vision system presented in this paper builds on our previous work [34], which describes an effective particle filter-based approach for predictive tracking of futsal players in scenes monitored by a single stationary camera, with (1) an updated and more comprehensive discussion of related work, (2) the description of all modules and capabilities of the proposed system, (3) improvements on the players detection algorithm, which now explores players appearance based on color information and uses the players' feet positions to specify their locations on the court, (4) a new set of experiments on an official match dataset and (5) a detailed analysis of the system's performance.

The remainder of this paper is organized as follows. Section 2 presents the related work. Section 3 introduces some fundamentals of our approach. Section 4 covers the proposed system. Experimental results are presented in Section 5, followed by the concluding remarks in Section 6.

2 Related Work

Several works have explored tactical and physical analyses in sport. The technologies used for detection and tracking, the core steps for those analyses, can be divided in two categories, as proposed in [39]: (i) *intrusive*, in which wireless sensors and tags are placed on players and ball; and (ii) *non-intrusive*, in which there are no extra objects placed in the game participants. In the next sections, we briefly describe those technologies and review the state-of-the-art.

2.1 Intrusive Systems

As intrusive systems make use of wireless tags and sensors, they are sensitive to signal collisions and interferences. The signals must be strong enough to be detected by antennas positioned around the game region, and tags must be light and small enough to allow players to perform comfortably and efficiently [39]. However, intrusive systems identify an object of interest, among a set of similar objects, in a precise and fast way, thus minimizing identity switches [28].

Different technologies can be used to detect and track players on this kind of systems. Microwave [9] and Radio Frequency Identification (RFID) [28] approaches use triangulation and arrival times of signals to estimate player position. In the Local Position Measurement (LPM) approach [3], the RFID sensors also emit microwaves to transmit performance data (e.g. position, speed, distance traveled) to a set of base stations. Another intrusive technology used for tracking is Ultra Wide Band (UWB) [7]. Its use can be advantageous in situations where there is no line of sight, and it determines player location based on Time Difference of Arrival (TDOA) and Angle of Arrival (AOA) [39]. Finally, there are some works that make use of Global Navigation Satellite System (GNSS) [1] and Global Position System (GPS) [45] to track players. GPS approaches, however, are not commonly used on indoor sports, as they can not, in most cases, estimate player position precisely and have poor efficiency levels in such environments.

2.2 Non-intrusive Systems

Non-intrusive systems are usually based on either infrared or computer vision techniques that use video cameras strategically positioned around the game environment [39,16]. This solution is widely adopted because it does not interfere in game action with extra apparatus inclusion and can ally robustness, confidence and performance levels.

In the field of detection and tracking of players for tactical and physical analyses, different image sources can be used. Some works are based on fixed cameras [30,10,11,24, 18,20,32,17,46,38], since they can capture, in most cases, all players actions in the game region, while moving cameras or broadcast images, on the other hand, can not always view all players in the scene during the entire game, which causes the loss of certain actions. Nevertheless, many works in that field use these types of image sources [22,35,49,15, 23,31], as they are most likely easier to obtain.

For the detection step, different techniques can be applied. Some papers use segmentation and morphological operations to detect players. In [13,22,17,15,46], a model that represent the game region is built based on color information, that is, the predominant game region color is used to create a color model of the background, allowing the extraction of regions that contain players. Similarly, the authors in [38,31] use histograms and color distributions to build a model that represents the player and it is used on the segmentation process. This model is built from players samples manually collected in a training phase. The detection step using color-based techniques is usually fast, but it is very sensitive to illumination variations, which may reduce their robustness and precision. Moreover, the techniques in [38, 31] demand on a laborious training step.

Adaptive background subtraction methods based on mixture of gaussians, in turn, are more resistant to illumination variations [50,40]. On the other hand, they are slower than color-based methods, and if the target stays static for sufficient time, it can be incorporated by the background model and, consequently, not be detected.

Other works adopt non-supervised training techniques, as an initial step of detection. One of the most explored is the training based on the use of Haar cascade classifiers [43]. In [30], the samples are manually extracted from game scenes, while in [26] the samples are automatically obtained from color-based segmentation. With the trained classifier, it is possible to detect players instances in images effectively. Unfortunately, the training phase has a high complexity cost and demands on a large number of samples. Furthermore, the detection process with this method is slow and it is more appropriate for post-game offline analyses.

Probabilistic approaches may be used with some of the previously described techniques to improve the detection, either in a multi-camera [30,10,11,47] or in a single-camera setup [19]. In a multi-camera setup, the authors in [30], for example, combine the detections in images from each camera made by Haar detectors in a multiple-hypothesis function. That function represents the likelihood of a player to be found in a certain court position, and it is built through the projection of the player location image coordinates in a virtual court plane. In [47], in turn, the authors present a multiple object tracking method based on a Bayesian formulation that uses the Reversible Jump Markov Chain Monte Carlo (RJ-MCMC) method. Target creation and removal are directly integrated into the probabilistic tracking framework. However, that approach requires global scene likelihood models involving a fixed number of observations (independent from the number of objects), which are sometimes diffi-

cult to obtain. On the other hand, the authors in [10,11] use background subtraction together with the Probabilistic Occupancy Map (POM) [18] technique to detect the players in different situations. In a single-camera setup, the authors in [19] use likelihood maps to estimate the locations of players based on their color distributions.

Regarding the visual tracking of multiple players, different approaches can also be found in the literature [36, 47]. The use of predictive filters is widely adopted in many papers [8]. In [46,13], Kalman filter is used to estimate the speed and position of players and link their trajectories. However, Kalman Filter is not adequate for multiple-hypothesis processes. For this reason, many authors choose to use Particle Filter for players tracking [34,30,24,49,15]. The complexity cost, in that case, increases proportionally with the numbers of players tracked. The players tracking approach of the vision system proposed in this paper belongs to this group of particle filter-based solutions.

Another approach on multiple players tracking is graph-based multiple-hypothesis and trajectory analyses. Graphs that represent the possible players' trajectories are built, modeling their positions in a given instant of time along with their transitions between frames [32,41,20,35,11,10]. The trajectories of players are searched in the graph using a similarity measure [32,41], K-Shortest Paths and linear programming [11], multi-commodity network flow [10] or modeled as a minimum edge cover problem [20,35]. Graph-based methods commonly have a high complexity cost and it is difficult to achieve real-time results in those approaches.

Finally, it is worth mentioning some popular commercial non-intrusive systems that can be found. One widely adopted by many teams in different sports is SportVU [5], that uses a set of fixed cameras installed around the game region and computer vision to track players in real-time. Other commercial vision-based tools can also provide this sort of information (e.g. StatDNA [6], Dartfish [2] and Opta [4]), but they focus on post-game analyses and some operators are responsible to extract the data manually.

3 Fundamentals

In this section, we briefly introduce some fundamentals to better understand our system, which uses a Bayesian filtering approach based on the so-called particle filter [12] to track players in a match.

Bayesian theory is a branch of probability theory that allows people to model the uncertainty about the world and the outcomes of interest by incorporating prior knowledge and observational evidence [14]. In other words, Bayes theorem is a mechanism for updating knowledge about some target state in the light of extra information from new data.

In the Bayesian approach to dynamic state estimation, one attempts to construct the posterior probability density function (pdf) of the state based on all available information, including the set of received measurements, considering the probability as a conditional measure of uncertainty [8].

From a Bayesian perspective, the tracking problem can be reduced as to recursively calculate some degree of belief in a state $\mathbf{s}_t$ at time t, taking different values, given the measured data $\mathbf{z}_i$, for $i = 1,...,t$. This way, Bayesian approaches assume the dynamic system is Markov - that is, the current state variable $\mathbf{s}_t$ contains all relevant information [12]. The measurement part of the Bayesian formulation is given by the likelihood function.

The problems in which an estimate is required every time a measurement is received can be convenient solved by applying a recursive filtering approach, where the received data can be processed sequentially [8]. To estimate present state based on past data, Bayes filters use a conditional probability $P(\mathbf{s}_t|\mathbf{s}_{t-1})$ to describe the system dynamics, that is, how the system's state changes over time. In location estimation, this conditional probability is the motion model - where the object might be at time t, given that it was previously at a specific location at state $\mathbf{s}_{t-1}$ [12].

4 The Proposed System

This section describes the proposed system to support tactical and physical analyses of futsal teams, which is divided in five main modules, as illustrated in Fig. 1. The first module performs the acquisition of image sequences of the game by using a single stationary camera. The second module is responsible to detect players in the game region, what is achieved by using an adaptive background subtraction method based on a mixture of gaussians [34,50] and on geometric constraints to check blobs sizes in the resulted binary image. In the third module, the proposed system performs the tracking of players, by using particle filters [34]. Specifically, each player detected is tracked by a separate particle filter. In addition, by using the third module, an operator can: (1) make the initial identification of each player, (2) manage identity switches involving different trackers, (3) re-identify a specific player, whose tracking has been interrupted and (4) start or stop the statistics computation. The fourth module, in turn, computes and stores tactical and physical data about the players, such as, court occupancy heat map, mean and maximum speeds and the distance covered by each one of the athletes. Finally, the fifth module consists in the end user interface of the system, in which the technical staff of a team may formulate their queries. The five steps aforementioned are described in the following.

4.1 Image Acquisition

The first module is responsible to acquire image sequences of the game. Image acquisition architectures differ on the number of cameras and in how they are located on the sports

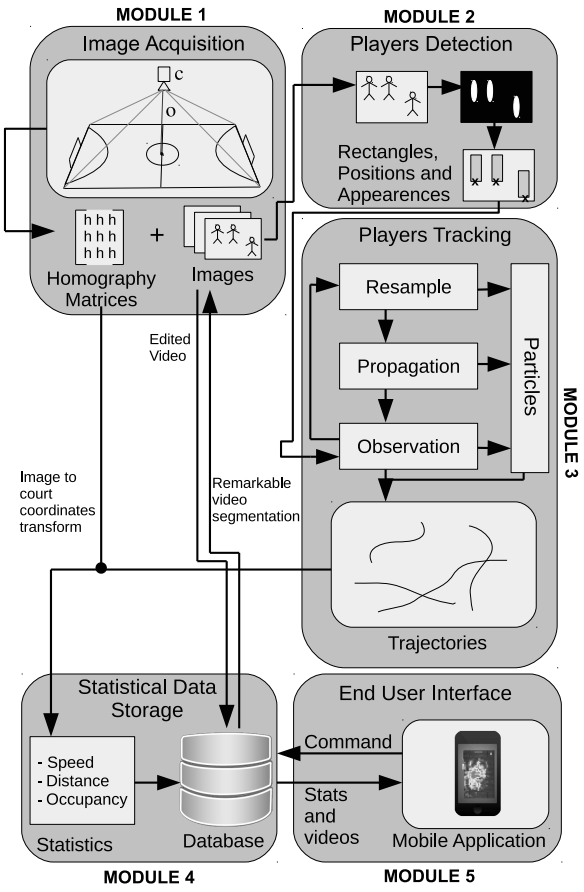


Fig. 1 Overview of the proposed system.

venue. Multiple fixed cameras allow to cover all the field of view and may ease the detection and tracking methods to be applied afterwards. However, those setups demand on sophisticated algorithms to solve the data association problem, in which we determine the correspondence between targets in different cameras. Moreover, the transmitting video data can require lots of bandwidth and will probably be more expensive than a single-camera system because of the extra sensors and the supporting infrastructure demanded. Considering those drawbacks of multiple-camera setups, we propose a solution based on the usage of a single stationary camera, c, placed in such a way it can capture top-view images of the court, as illustrated in Fig. 1. With this setup we reduce the effects of occlusions among players.

The camera used is the PROSILICA GC750 with Gigabit Ethernet interface. Images are acquired at 30 frames per second with dimensions of 752×480 pixels. To capture the entire court, we make use of a COMPUTAR manual varifocal wide angle lens with focal length ranging from 1.8 to 3.6 mm (model T2Z1816CS). Unfortunately, this kind of lens causes substantial undesirable spherical distortion on the images, as illustrated in Fig. 2(a).

In order to reduce the negative effects of such distortion, the camera is calibrated with the algorithm proposed by Zhang [48], so that its intrinsic parameters are estimated and used to undistort the images. Additionally, as only the court area contains relevant information, the undistorted image is adequately cropped to represent the region of interest, thus reducing the amount of pixels to be processed and resulting in a court image as the one illustrated in Fig. 2(b).

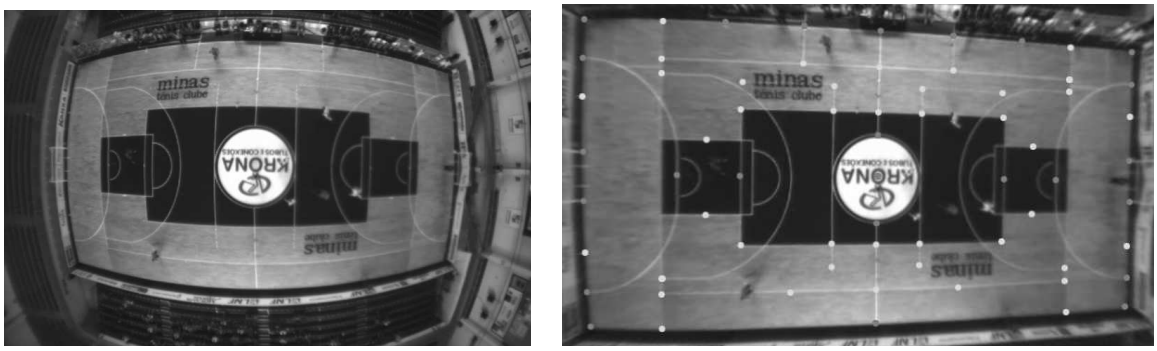
Once the camera pose has been properly determined, the image acquisition module also computes two different homographies H_1 and H_2. The homography H_1 is used to map points in the camera's image plane to their corresponding ones in the court area, which is essentially a plane in the world coordinate system. Based on H_1, the system estimates the distances traveled by the players and their speeds. The homography H_2, in turn, maps points in the camera's image plane to their corresponding ones in a virtual plane, as shown in Fig. 7(a). Therefore, H_2 is responsible to support the creation of a player's heat map, which is an indicator of his/her presence in different parts of the court. The map gets heated up in areas where the player has had more control of the ball and does most of his/her work, i.e it turns redder as the player's presence in a particular area increases. To estimate H_1, we identified a set of 67 static scene points in the court (e.g. the center of the court, its corners, among others) whose positions are known in a world-coordinate system and find their corresponding positions in pixels in the camera's image plane. All those points are shown in green in Fig. 2(b). On the other hand, to estimate H_2, we have used a subset of 25 points, which are shown in red in Fig. 2(b) and can also be found in the virtual plane illustrated in Fig. 7(a).

4.2 Players Detection

The second module of the proposed system performs the detection of players in the game region. To achieve this goal, we use an adaptive background subtraction method based on Gaussian Mixture Models (GMM), as proposed in [50]. Frequently, the scene background presents some regular behavior that can be described by a model. With this model, it is possible to detect a moving object by searching image pixels that does not fit the model.

In our solution, the background model B is estimated from a training set denoted as S. This training set is built from pixels values $\mathbf{x}$ sampled over a time period ΔT, so that at time t we have $S_{\Delta T} = \{\mathbf{x}_t, \mathbf{x}_{t-1}, ..., \mathbf{x}_{t-\Delta T}\}$, and the estimated background model is denoted by $\hat{p}(\mathbf{x}|S,B)$ [50].

Over time, each new sample is incorporated to the set and the old ones are discarded, so that the model is updated in order to adapt to changes in the scene. In the recent samples, however, there could be some values that belong not only to background but also to foreground objects, represented by the foreground model F. This estimative should



(a) (b)

Fig. 2 (a) Example of court image captured by our system, suffering from spherical distortion. (b) Static scene points used to estimate H_1 and H_2.

be denoted by $p(\mathbf{x}_t|S_{\Delta T},B+F)$ and in a GMM with M components, is given by [50]:

$$\hat{p}(\mathbf{x}|S_{\Delta T},B+F)=\sum_{j=1}^{M}\hat{\pi}_j\cdot\mathcal{N}(\mathbf{x};\hat{\mu}_j,\hat{\sigma}_j^2 I), \tag{1}$$

in which $\hat{\mu}_j$ is the estimate of the mean of the j^{th} Gaussian component and $\hat{\sigma}_j^2$ is the estimate of the variances that describe the j^{th} Gaussian component. The covariance matrices are assumed to be diagonal and I, the identity matrix, has proper dimensions [50]. The mixing weights (the portion of data accounted by the j^{th} Gaussian), denoted by $\hat{\pi}_j$, are non-negative and normalized in such a way they sum to one. The weight $\hat{\pi}_j$ may be considered as the probability of a sample being derived from the j^{th} Gaussian component. In other words, the weight $\hat{\pi}_j$ specifies the amount of time that certain intensity values (and, similarly, color values) are captured from the scene. This means that the estimation of Gaussian components that correspond to background colors is based on the persistence and variance of each component of the mixture. The likely background colors are those that are captured from the scene during longer periods of time and present a more stable behavior [40].

To estimate the background model from the mixture, the algorithm assumes that Gaussian components having the most supporting evidence and the least variance are most likely to be part of the background. To determine those components, our approach considers that the background contains L most likely intensity values. In a clustering approach, static objects tend to form large and concise clusters of pixels with the same value, while moving ones tend to form sparse clusters. This way, the intruding foreground objects will be represented, in general, by some additional clusters with small weights $\hat{\pi}_j$ [50]. The background model can be approximated by the first L largest clusters:

$$p(\mathbf{x}|S_{\Delta T},B)\sim\sum_{j=1}^{L}\hat{\pi}_j\cdot\mathcal{N}(\mathbf{x};\hat{\mu}_j,\hat{\sigma}_j^2 I). \tag{2}$$

Sorting the components by their weights $\hat{\pi}_j$ in descending order, we obtain:

$$L=\arg\min_{l}\left(\sum_{j=1}^{l}\hat{\pi}_j>(1-\beta)\right), \tag{3}$$

in which β is a measure of the maximum portion of data that can belong to foreground objects without influencing the background model [50]. This way, the first L of the ranked components whose weights exceed $(1-\beta)$ are deemed to be the background.

A limitation present in earlier background subtraction based on GMM approaches was caused by the use of a fixed number of Gaussian components for each pixel over the time. To increase the accuracy and reduce computational cost, the technique in [50] applies an online procedure to constantly update not only the GMM parameters but also the number of components to be used. Given a new data sample $\mathbf{x}_t$ at time t, the recursive update equations are:

$$\hat{\pi}_j\leftarrow\hat{\pi}_j+\alpha\cdot(o_j^t-\hat{\pi}_j)-\alpha\cdot\rho, \tag{4}$$

$$\hat{\mu}_j\leftarrow\hat{\mu}_j+o_j^t\cdot(\alpha/\hat{\pi}_j)\cdot\delta_j, \tag{5}$$

$$\hat{\sigma}_j^2\leftarrow\hat{\sigma}_j^2+o_j^t\cdot(\alpha/\hat{\pi}_j)\cdot(\delta_j^{\Delta T}\cdot\delta_j-\hat{\sigma}_j^2), \tag{6}$$

in which $\delta_j=\mathbf{x}_t-\hat{\mu}_j$. The constant α describes an exponentially decaying envelope, used to limit the influence of the old samples and, approximately, $\alpha=1/\Delta T$. For a new sample, the ownership o_j^t is set to 1 for the "close" component with the largest weight $\hat{\pi}_j$ and the others are set to zero. A sample is said "close" to a component if the Mahalanobis distance from the component is for example less than three standard deviations. The squared distance from the j^{th} component is calculated as $D_j^2(\mathbf{x}_t)=\delta_j^{\Delta T}\cdot\delta_j/\hat{\sigma}_j^2$. If there are no "close" components, a new component is generated with $\hat{\pi}_{M+1}=\alpha$, $\hat{\mu}_{M+1}=\mathbf{x}_t$ and $\hat{\sigma^2}_{M+1}=\sigma_0^2$, in which σ_0^2 is some initial variance with appropriate value [50]. If the maximum number of components is reached, the component with the smallest weight is discarded. Finally, ρ is

the negative Dirichlet prior weight, which will suppress the components that are not supported by the data. If a component has negative weights, it is discarded. After each update, the weights are again normalized.

At the beginning of the execution, the GMM is started with one component centered on the first sample. New components are added or discarded as aforementioned, so that the number of components is dynamically updated and the background model is effectively estimated. Pixel values that do not fit the model are thus considered as belonging to the foreground, until there is a Gaussian component with enough evidence to support their inclusions in the background. In the detection process, only regions inside the court area are considered, to avoid that the movement of coaches, referees or even supporters lead to wrong detections.

To increase robustness, it is necessary to detect moving shadows pixels upon pixels labeled as foreground. In the background subtraction process, a pixel is detected as shadow if it is considered as a darker version of the background, defined by a threshold τ. As shadows pixels are marked with a specific value in the resulted image (127 in the present case, resulting in grey pixels), they can be easily removed with a simple threshold operation. Then, we have a binary image in which black pixels represent the background and white pixels represent foreground objects.

The second step of our player detection approach is to perform some morphological operations, as opening (to remove noise pixels and small objects) and closing (to remove small holes on foreground blobs). At this moment, bounding rectangles are assigned to each blob as possible players locations creating a set R of regions of interest.

All regions in set R must be checked against some geometrical constraints, to verify if they really correspond to players, given their respective width and height and their positions. The i-th region in R is discarded if $w_i < w_{min}$ or $h_i < h_{min}$, in which w_i and h_i denote the width and height of the i-th region, respectively, and w_{min} and h_{min} are the minimum values for width and height that a region may assume to represent a potential player in the scene. Similarly, our approach evaluates if $w_i > w_{max}$ or $h_i > h_{max}$, in which w_{max} and h_{max} are the maximum values for width and height for a region that may represent a player. In those cases, if $w_i > w_{max}$ or $h_i > h_{max}$, the approach recursively splits the region into smaller rectangles until they meet the dimensions constraints and, in the following, updates the set R.

To process only detections that are inside the court area, the proposed approach considers the players' feet positions instead of the centroids of their corresponding regions in R. By doing so, we have increased the accuracy of our system to determine the players' locations on the court. Our key observation is that the perspective projection effect, especially in regions that are distant from the image center, makes the centroid of a detected player an unreliable cue about its actual location, as illustrated in Fig. 3(a). In contrast, by assuming the player's location is given by the position of the point between his/her feet, our approach can deal with particular situations, as for example, when a player runs near the boundaries of the court or when he/she is performing a throw-in. Usually, in those situations, differently from the player's feet, the centroid is outside the court area.

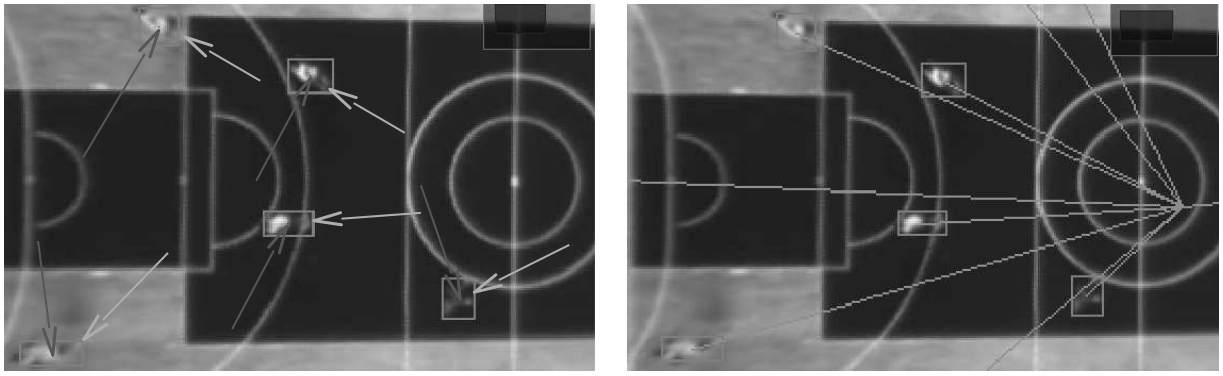
To estimate the location of a player from his/her feet positions, the centroid of his/her corresponding region is connected through a line segment to the image center. Specifically, the intersection point between such a line segment and the bounding rectangle of a player is considered as his/her location estimated from his/her feet, as illustrated in Fig. 3(b). If the line segment does not intersect the bounding rectangle, that is, the line segment is completely inside the rectangle, the centroid of the region is then considered as the estimate of the player's location. Thus, given the estimated locations of players in the camera's image plane, the homography H_1 is used to compute their corresponding ones in the court area and check if the detected objects are inside the limits of the court. If a specific detection is not inside the boundaries of the court, it is discarded.

The final step of the detection module consists in to compute an appearance model that captures the color information of the player and is later used by the tracking module. The appearance model consists of 3 normalized color histograms, with 16 bins each, which are built in the HSV space (one histogram for each channel).

4.3 Players Tracking

The third module performs the tracking of players, that is, it estimates players positions at a given time and link their detections over successive frames. In this work, we make use of Particle Filter for this task [34]. Particle filter is a predictive filter, which uses information from the present state of an object to infer its state in the next instant of time. To make this possible, the filter uses a motion model, which describes the motion dynamics of the objects. Through this model, the filter can make a prediction of the position of the object in the next instant of time, which is corrected by an observation model (e.g. the position of the detected player), since it is not exactly known how the object is moving at that moment. With this adjustment, we minimize the effects of accumulated errors that can lead to erroneous predictions. If it is not possible to directly observe the object (for example, in a miss detection), the filter uses only the prediction to keep tracking it until the object can be detected again.

To work with multimodal functions, as in the present case, the filter models its probability functions using a set of N samples, or particles – hence the origin of its name. Each particle i has at a time instant t a state $\mathbf{s}_t^i$, which contains an information that represents the player. Each particle has also a weight w_t^i, which refers to how good that sample



(a) (b)

Fig. 3 (a) Centroids and feet locations of players: blue arrows indicate the centroids locations, while green arrows indicate the feet locations. (b) Lines connecting the centroids of detected regions to the image center.

is, or, in other words, what is the likelihood of the player to be found in that position if an observation is made at that instant. In the proposed work, we use a vector with four variables to model the state of a player, so $\mathbf{s}_t^i = [x\ y\ v_x\ v_y]^T$, in which the first two variables are the position in 2D space, v_x is the velocity on x axis and v_y is the velocity on y axis. The estimated state of the player tracked, $\hat{\mathbf{s}}_t$, is given by:

$$\hat{\mathbf{s}}_t = \sum_{i=1}^{N} w_t^i \cdot \mathbf{s}_t^i. \tag{7}$$

We start tracking a player from its first detection. To each new detection is associated a tracker, consisting of its own particle filter, the player identification, his/her position history and his/her appearance history. The tracker is considered "valid" if the player associated to it is detected on a minimum number of frames, denoted by γ_{min}. By "valid" we mean that the tracker can compute the trajectories of the player, to avoid computing trajectories of some tracked objects generated by noise detection. In the same way, a tracker can only live without an associated detection for a limit number of frames, γ_{lim}, being removed after that.

When a new tracker is created at time t_0, a set of N particles is created with states $\mathbf{s}_{t_0}^i = [x\ y\ v_x\ v_y]^T$. The values of x and y are computed according to a normal distribution around the position of the associated detection with variance $\sigma^2_{x,y}$. Moreover, v_x and v_y are initialized with values equal to zero. All particles have the same weight, that is $w_{t_0}^i = 1/N$.

Next, an iterative process begins, which is repeated for every new frame, consisting of the resample, propagation and observation phases. In the following, we describe each one of those phases, considering t as the current time.

4.3.1 Resample

In this phase, particles are resampled according to their weights in order to build a new set with N samples based on the previous one. In a $[0,1]$ closed interval, we map portions of this interval to each one of the particles, in such a way that those with larger weights receive larger portions. We then generate a random number n and we choose the particle that has the interval which contains n. This way, we benefit particles with larger weights, but we still admit repetitions and also allow small weight particles to be selected.

4.3.2 Propagation

In this phase, we propagate the particle set by using the motion model to build the estimate of state $\mathbf{s}_{t+1}$. That is, we basically make a prediction of the next state. We employ the constant velocity motion model in this work, as proposed by the authors in [30], motivated by the fact that the variations between frames are very small when images are captured at 30 frames per second. Because of that, this model is able to manage sudden changes of directions of the players motion, including those movements that depend on the ball position in the field. In this model, we have:

$$(x,y)_{t+1} = (x,y)_t + (v_x,v_y)_t \cdot \Delta t, \tag{8}$$

$$(v_x,v_y)_{t+1} = (v_x,v_y)_t, \tag{9}$$

in which Δt is the time step. However, as the particle filter deals with the likelihood of an event, there are uncertainties that should be considered. Those uncertainties can be seen as the process noise and we model it as random errors from a zero mean normal distribution with variance $\sigma^2_{v_x,v_y}$, which is empirically defined. Such errors help differentiate the state of repeated particles, improve the representativeness in that point and avoid repetitions that break the tracking step [30].

With a time step $\Delta t = 1/30$, the model can be rewritten in matrix terms as:

$$\mathbf{s}_{t+1} = \begin{bmatrix} 1 & 0 & 1/30 & 0 \\ 0 & 1 & 0 & 1/30 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \left(\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_t + \begin{bmatrix} 0 \\ 0 \\ e_{v_x} \\ e_{v_y} \end{bmatrix} \right), \tag{10}$$

in which e_{v_x} and e_{v_y} are the process noise.

4.3.3 Observation

In this phase, the estimates are adjusted by an observation model $\mathbf{z}$ of the object, to confirm or correct them. At this moment, we compute the new particle weight, which denotes how good that representation is. In other words, what is the likelihood of the player be found in that position if an observation is made at that moment, denoted by $P(\mathbf{s}_{t+1}|\mathbf{z}_{t+1})$. As we are tracking players and estimating their positions, we adopt a model in which $\mathbf{z}_{t+1} = [x\ y]^\top{}_{t+1}$, so only the position information is considered.

As we deal with multi-player tracking, it is necessary to decide which detection should guide each tracker, to adjust its prediction in this phase. This way, each tracker should be associated with one detection at most and this problem is called an association problem. To solve it, we apply the well-known Hungarian Algorithm [25]. This technique calculates the cost of all association possibilities, given in our work by the Normalized Euclidean Distance between a position of one detection and the predicted position of a tracker. The algorithm makes the appropriate associations in such a way that each tracker is associated with one detection at most with the smallest possible cost, in polynomial time.

With all associations made, we check if each cost given is smaller then a threshold λ, which controls the maximum acceptable cost. We do this to minimize unreal situations, most likely caused by false positive detections (e.g. a player that is detected at the penalty mark in one frame and in the next he/she is detected at the center of the court, being impossible to a human to travel such distance in such a small period of time). If a detection that does not really exist is associated to a tracker, this tracker uses only its prediction and appearance data to correct its estimate. When the number of strikes is larger than γ_{lim}, the tracker does not have detections associated to it for a substantial number of frames and then we remove it. On the other hand, if a valid detection is associated to the tracker, we use it to adjust the prediction.

To estimate the particles weights, two different methods are used, which are chosen in accordance with the existence or not of a detection associated to the tracker. Considering such an existence, we compute the Euclidean Distance d between the position (x,y) of the i-th particle at state $\mathbf{s}^i_{t+1}$ and the player's location according to his/her corresponding detection. We use d in a normal probability density function that returns the particle weight and is given by:

$$\frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{\left(-\frac{d^2}{2\sigma^2}\right)}, \tag{11}$$

in which $\sigma^2 = \sigma^2_{x,y}$. We set the initial variance $\sigma^2_{x,y}$ for the position based on the mean size of the player in images. During tracking, that variance is inversely proportional to the number of successfully tracked frames for a player (down to a lower limit θ_l). Hence, the longer a player is tracked successfully, the less the particles are spread. In the same way, when it is not possible to detect the player associated to a tracker, we increase the variance up to a higher limit θ_h, to spread the particles and to make better estimates.

On the other hand, if there is no detection associated to the tracker, the particle weight is computed by a four-step method that combines prediction and appearance data. The first step consists in to estimate the particle's appearance model, which similarly to the player's appearance model, consists of 3 normalized color histograms, with 16 bins each, which are built in the HSV space (one histogram for each dimension). In a second step, the method computes the similarities between the estimated particle's appearance model and all the other models previously computed by the tracker. The largest similarity value obtained is returned. To compute the similarity between two models, the method uses histogram correlation. In this case, the values 0 and 1 represent the smallest and largest similarities, respectively. Therefore, analogously to the strategy used in [30], by considering an appearance model A as a list of 3 histograms and A_i as the i-th histogram in this list, we compute the similarity between two appearance models A_1 and A_2 as follows:

$$f_s(A_1,A_2) = \prod_{i=1}^{3} f_c(A^i{}_1, A^i{}_2), \tag{12}$$

in which $f_s(\cdot)$ is the similarity measure between two models and $f_c(\cdot)$ is the correlation function between the counting values obtained from the histograms bins. The largest similarity value obtained, ${f_s}^{max}$, is used in a probability density function given by $5 \cdot ({f_s}^{max})^5$, which is empirically determined, in such a way that particles that are less similar to the tracker have smaller weights, while particles that are more similar have larger weights. The third step, in turn, is analogous to the scenario when a detection is associated to the tracker. However, the Euclidean Distance d is now computed between the position (x,y) of the i-th particle and the mean position of the estimate in the propagation phase. The fourth and final step consists in to perform a weighted sum to combine the weights obtained in the second and third steps. Thus, the particle weight for the case when there is no detection associated to the tracker is given by:

$$w_{sum} \cdot w_s + (1 - w_{sum}) \cdot w_d, \tag{13}$$

in which w_s is the weight of the similarity obtained in the second step, w_d is the weight of the Euclidean distance obtained in the third step and w_{sum} is a factor that determines the contribution of each one of the variables in the composition of the particle's weight.

Afterwards, we normalize the weights so that they sum up to one. From that, the estimated state $\hat{\mathbf{s}}_{t+1}$ of the tracker

may be computed again by using Equation (7) and the particle filter is ready for a new cycle. For each iteration, our approach stores in the tracker history the location of the player, given by its estimated state, so that we can compute his/her trajectories over time, as well as his/her statistics of interest.

Importantly, an operator manually performs the initial identification of each player to be tracked through the players tracking module. The operator sets the tracker's identifier with the player's number and informs the team he/she belongs to. The operator is also responsible for undoing complex confusing situations during the game as, for example, cases of identity switches involving different trackers. In addition, the operator can perform the re-identification of a specific player who has been tracked, but had his tracking interrupted by any reason. Finally, the operator is responsible to start and stop the statistics computation, in order to avoid that interruptions during the game can affect and damage the physical and tactical data computed.

4.4 Statistical Data Storage

Once the players trajectories have been obtained, the fourth module performs the computation of players statistics. The statistical data estimated consist of occupancy heat maps (tactical data), as well as distance covered, mean and maximum speeds of players (physical data).

To estimate the player's heat map, the corresponding tracker holds an occupancy matrix with the same dimensions of the virtual court illustrated in Fig. 7(a). At each time instant t, that is, at each frame processed, our approach estimates the player's location in the image plane, which is then mapped to the virtual court by using the homography H_2. Next, our approach increments the occupancy matrix position that corresponds to the mapped location. When the tracking of a player is ended, the occupancy matrix stores the court positions that have been visited and the number of times that the player was present in those positions. Finally, by using a colormap and the aforementioned occupancy matrix, the player's heat map is obtained and stored.

On the other hand, differently from the heat map computation, our approach computes the physical data of a player only at a rate of 15 frames per second. This sampling rate is adopted because of two main reasons. Firstly, note that our solution captures images at 30 frames per second. At this frame rate, the distances traveled by the athletes and, consequently, their speeds do not vary significantly between consecutive frames. Therefore, to estimate physical data at every time instant creates an unnecessary computational cost. Secondly, the estimated location of a player may sometimes modify significantly between frames, even when the player has moved himself/herself very little. This occurs because of the size variation of the bounding rectangle assigned to the blob that potentially represents the player. In this case, the player's location estimated from his/her feet may alter in a significant manner and, thus, add errors to the statistics computed. For those reasons, we have performed a careful analysis about the most appropriate sampling rate to be used. This analysis was based on a statistical test, which is described in the experimental results section, and concludes that a rate of 15 frames per second was adequate for our application.

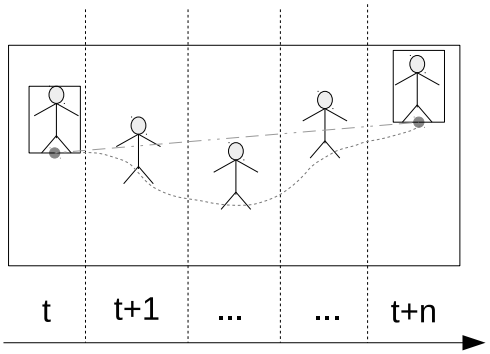


Fig. 4 Example of a tracking interruption event of a specific player.

Basically, for every 15 frames, the current and immediately preceding locations of the player are mapped to the court area by using the homography H_1. Next, the Euclidean distance is calculated between those locations. The value obtained is added to the variable of the tracker responsible for storing the distance covered. By using the distance information, in turn, the player's speed is derived. The maximal speed of a player is given by the largest "current" speed estimate. Such estimate of speed is computed by dividing the Euclidean distance obtained at this stage by the period of time between the computations (0.5 seconds, since the statistics are calculated every 15 frames). Finally, the player's mean speed is obtained by dividing his/her total distance traveled by the period of time that he/she has been tracked.

The players statistics are stored in data structures at predetermined periodic time intervals or just before a tracker is removed. Unlike the trackers, which are volatile entities, those data structures can permanently keep the tactical and physical data of the players. The information stored in the data structures may come from different trackers. Those data structures are still used in some special recurrent situations, as the one described in the following.

Consider a tracking interruption event at time instant t of a specific player, as illustrated in Fig. 4, where a player must be re-tracked. In that scenario, the player keeps moving along the path indicated by the dotted blue line without being detected by our system. Suppose now that at a posterior time instant the player has been detected and, consequently, tracked again. By observing such a situation at time $t+n$, the system operator re-identifies the player and, in this case, the data structure containing his/her statistics are retrieved and copied to the new tracker created. Thus, the player's tracking and statistics computation can be properly continued.

To minimize errors in the computation of players physical data, caused by tracking interruption events as the one illustrated in Fig. 4, we calculate the Euclidean Distance between the last estimated player's location (at time instant t) and the new location where he/she has been reindentified (at time instant $t+n$). That distance is represented by the red dashed line in Fig. 4 and is used together with the information about the number of frames without detection ($n = (t+n) - t$) to update the player's statistics.

Another situation that is addressed by the statistical data storage module is the substitution of a player. In futsal, a player who has been substituted by another one may later return to the game, if necessary. When a player leaves the court area to be substituted, he/she is not any more detected by our system. In this case, his/her statistics are properly stored in the corresponding data structure before the tracker is removed. Posteriorly, when that player returns to the game, the system operator performs his/her re-identification and activates a boolean variable, which specifies that such a player has already participated in the game. Thus, the data structure containing his/her statistics are retrieved and copied to the new tracker created. To proceed with the computation of physical data of a player who has been substituted, our solution consider the last estimated player's location (before leaving the court) as his/her current one.

All the statistical data computed by our solution are stored in a database, which receives requests for information of end-users from a specific mobile application that is briefly described in the next section.

4.5 End User Interface

The fifth and last module consists in the end user interface. Basically, through this module, which has been specially developed to run on Android mobile devices, the technical staffs of futsal teams formulate their queries regarding the players statistics and visualize the results. The Android platform has been chosen due to its popularity, open-source nature for an extensive customisation and low cost. The developed application retrieves information from the system's database by using the open standard format named JavaScript Object Notation, which is frequently considered as an effective and lightweight data-interchange format.

The information requested is presented on different screens. At the beginning of the application's execution, the first screen presented to the end user is the list of teams. When the user selects a specific one, a screen containing its corresponding players is shown. By clicking on a player, his/her information is provided, such as name, number, position and physical statistics collected. An additional button allows the visualization of the player's heat map as well.

By using the application, the user can also obtain video segments that capture actions of interest for the analyses of the technical staff. Specifically, the user may run a command at the moment the key event occurs. The system can then edit a video file containing some previous and subsequent game actions along with the one of interest and make them available for streaming in the mobile device. The technical staff can show this video segment to players to correct team flaws or to show weaknesses of the opponent team. To achieve this goal, a video buffer is used by our system, whose size may be adapted according to the end user's demand.

Table 1 Parameters' values used in the experiments.

Parameter	Value
w_{min}, h_{min}	5 pixels
w_{max}	40 pixels
h_{max}	30 pixels
N	350 particles
$\gamma_{min}, \gamma_{lim}$	15 frames
$\sigma_{x,y}$	5 pixels
σ_{v_x,v_y}	5 pixels
θ_l	3 pixels
θ_h	7 pixels
λ	0.05
w_{sum}	0.70

5 Experimental Results

In order to evaluate the accuracy, efficiency and applicability of our system, we tested it on a challenging set of experiments. Those experiments allowed us to individually assess the proposed approaches for players detection, players tracking and players statistical data computation.

We have used two datasets for this task, namely, one dataset of an official futsal match and another one of a training match, which include different tactical and physical demands players may be exposed to. Both datasets come from the Minas Tênis Clube professional futsal team acting in its arena. The images were captured at 30 frames per second with dimensions of 752×480 pixels. However, as mentioned in Section 4.1, we crop the court region in the images, resulting in frames with dimensions of 640×370 pixels.

We have manually annotated the positions of the athletes in the images with a bounding box around each player, once every 15 frames, to create the ground truth for each dataset. The mean bounding box of a player in the images is 30×20 pixels. We chose the 15 frames time step because it would be a very laborious task to manually set up those boxes in every frame of the sequences. Similarly, the sequences lengths were chosen in such a way they could provide relevant information for the tests without preventing their executions.

As shown in Section 4, different parameters must be initially defined, so that our system can be properly used. A set up phase was conducted and the values of those parameters were empirically determined by using our knowledge about the problem (see Table 1). Next, the experimental results are presented.

5.1 Official Match Dataset

The first dataset consists of an image sequence captured from an official futsal match, which contains 12,870 frames and corresponds approximately to a 7 minute long video. Official matches demand on greater physical loads compared to training matches. Moreover, tactical approaches are applied and validated in real challenging game actions.

5.1.1 Players Detection Results

To evaluate the detection of players performed by our system, we use the MODA CLEAR metric [21], which stands for Multiple Object Detection Accuracy. This metric has become one of the standards for the evaluation of object detection algorithms in the computer vision area and utilizes the number of missed detections (false negatives) and false positive counts. We compute the number of false negatives (FN), false positives (FP) and true positives (TP) based on an overlap ratio between the annotated box in the ground truth and the bounding rectangle of the i-th region in R where the player was detected. For a given overlap ratio threshold τ_{ov}, a detection D is a true positive if [42]:

$$\frac{|D_i \cap G_i|}{|D_i \cup G_i|} \geq \tau_{ov}, \tag{14}$$

in which D_i and G_i are the i-th mapped pair of detection and ground truth.

The choice of the value τ_{ov} varies with the evaluation context. For larger objects, that cover several thousands of pixels, values such as 0.5 or 0.7 are suitable for the threshold. However, for small objects as in this work, in which players have a mean size of 30×20 pixels, even small deviations in size or position of the bounding box can induce significantly less overlap [42]. In order to demonstrate the impact of the overlap ratio threshold on the system's performance evaluation, we vary τ_{ov} value between 0.1 and 0.5.

As the MODA metric is originally defined for single frames, we can compute the Normalized MODA (N-MODA) for the entire sequence as [21]:

$$\text{N-MODA} = 1 - \frac{\sum_{t=1}^{N_{frames}} (c_{FN} \cdot FN_t + c_{FP} \cdot FP_t)}{\sum_{t=1}^{N_{frames}} N_{Gt}}, \tag{15}$$

in which FN_t is the number of false negatives, FP_t is the number of false positives and N_{Gt} is the number of objects on the ground truth (TP + FN), all these three values related to a given frame at time t. Thus, the false negative counts at that time is given by the number of objects on the ground truth for that frame minus the number of true positives found in that image. On the other hand, the false positives in a frame are calculated by subtracting the number of detected objects in that frame by the number of true positives obtained in the same image. The weights c_{FN} and c_{FP}, in turn, can be viewed as cost functions used to weigh the impact of false negatives and false positives, respectively. As in [21], in this assessment, c_{FN} and c_{FP} are both equal to one.

Table 2 shows our players detection results for the *Official Match* dataset. For clarity, we also calculate the F-Score as the harmonic mean between precision (Pre.) and recall (Rec.). Fig. 5 shows the impact of varying τ_{ov} value over the N-MODA and F-Score values and over the global mean error obtained. To calculate the global mean error, we firstly compute the mean error in each frame. For each detection regarded as true positive, we calculate the Euclidean distance between the foot position given by the bounding rectangle of the detection and the foot position given by the annotated box, both positions in court coordinates. Then, we calculate the mean distance for the frame and finally the global error is obtained by computing the average of the mean distances obtained in each frame of the sequence.

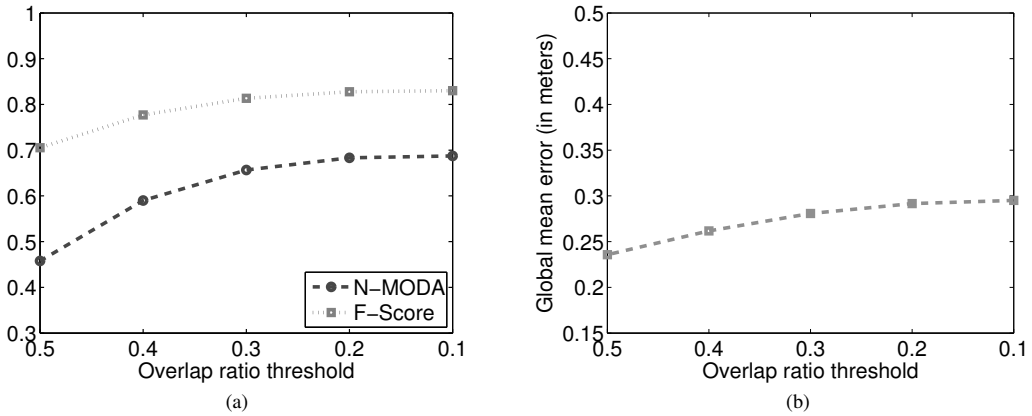
Our results show that, lowest values of τ_{ov} are more appropriate and lead to significant values of N-MODA and F-Score, as aforementioned, since the system handles small objects in the scene. However, upon decreasing the threshold value, there is an obviously increment on the error, since larger deviations are allowed for true positive detections. Those deviations are caused, generally, by our system merging the detections of two or more athletes or by partial detection of the players. Still, the global mean error for the players detection in this sequence is less than 30 cm, regardless of the overlap ratio threshold used. This is a promising value, considering the dimensions of the court (38×19 m).

Moreover, this dataset is very challenging and imposes several difficult situations in the detection process. A complex situation, for instance, is that substitute players and technical staff stay too close to the court area during the games in this arena. Specifically, the substitutes warm-up very close to the court boundaries and sometimes they even step into the court. Thus, they are detected, which increases the false positive counts. The same happens with coaches, when they pass instructions to their teams. To handle this, it is possible to narrow even more the region considered as the playing area. However, this restriction also leads to interruptions on the detection of athletes that are moving on the borders of the court or are performing a throw-in or a corner kick, increasing the false negative counts in those cases.

We decided to not detect and track the two referees in our system. In futsal, each referee moves sideways off the playing area, close to the sidelines. Considering them in this as-

Table 2 Players detection results for the *Official Match* dataset.

τ_{ov}	TP	FP	FN	Prec.	Rec.	F-Score	N-MODA
0.1	6563	655	2032	0.909	0.764	0.830	0.687
0.2	6545	673	2050	0.907	0.762	0.828	0.683
0.3	6431	787	2164	0.891	0.748	0.813	0.657
0.4	6144	1074	2451	0.851	0.715	0.777	0.590
0.5	5576	1642	3019	0.773	0.649	0.705	0.458



(a) (b)

Fig. 5 Players detection results for the *Official Match* dataset, considering different values of overlap ratio threshold (τ_{ov}). (a) N-MODA and F-Score values as functions of the overlap ratio threshold used. (b) Global mean errors obtained according to the threshold used.

sessment would result in an additional cost to annotate their positions and to check the data. However, the referees are also a cause of some false positives. As mentioned earlier, the arena where the sequences were recorded lacks on space to accommodate the presence of substitute players, technical staff and the referees on one side of the court. This way, very often one of the referees moves over the sideline or even inside the court, being detected in such cases. Moreover, when there is a foul or a player is injured, one of the referees (in some cases, both) enters into the court to indicate the location of the free-kick and to verify the athlete's situation. In those cases, the referee is also detected in several frames, which increases the false positive counts. False positives are, still, caused by the ball detection and the detection of some light shadows not filtered in the process.

On the other hand, a major part of the false negatives are caused by a large similarity between the floor of the court (the background) and players' uniforms. In some areas of the court, the appearance of the athletes is not discriminative enough to consider him/her as a foreground object. The images captured by our system's camera also introduce some difficulties in the process, since they are captured in low resolution and they appear blurred and have lots of noise in some regions, making it difficult to detect the players. Furthermore, there are also cases when two or more players are very close to each other, and they are considered as a single blob that generates only one valid detection, not being treated by the recursive splitting of the bounding region, as it still respects the geometric constraints. Nevertheless, the results obtained are encouraging and demonstrate the potential of our approach to locate players in the images.

5.1.2 Players Tracking Results

In order to assess the players tracking results on a sequence, we use the MOTA CLEAR metric [21], which stands for Multiple Object Tracking Accuracy. This metric is also widely applied in computer vision. Once more, we need to compute the number of false negatives, false positives, true positives and the number of identity switches for a given tracking ground truth. We consider that a track is a true positive if the Euclidean distance between the estimated position given by the tracker and the feet position defined by the annotated box is smaller than a certain threshold, τ_d. Similarly to the detection, we vary the value of τ_d between 0.50 meters and 1.75 meters, so that we can demonstrate the impact of this threshold in the tracking performance evaluation. The MOTA measure is given by [21]:

$$MOTA = 1 - \frac{\sum_{t=1}^{N_{frames}} (c_{FN} \cdot FN_t + c_{FP} \cdot FP_t + c_{ID} \cdot ID_t)}{\sum_{t=1}^{N_{frames}} N_{Gt}}, \tag{16}$$

in which c_{FN}, FN_t, c_{FP}, FP_t and N_{Gt} are the same as defined in Equation (15), and $c_{FN} = c_{FP} = 1$. However, the count of false negatives, false positives and true positives occurs differently. To increment the false negatives counts for a frame at time t, the player must exist in the ground truth at that time (i.e., he/she must be playing at that moment), but not has been tracked by an identified tracker. To increment the true positives counts for a frame at time t, the player must also exist in the ground truth at that time, has been tracked by an identified tracker and the Euclidean distance between his/her estimated position and his/her ground truth position must be smaller than τ_d. On the other hand, there are three different ways to increment the false positive counts for a frame at time t. The first one consists in the player to not exist in the ground truth at that time and he/she has been tracked by a previously identified tracker. In the second one, the athlete exists in the ground truth at time t and was tracked by an identified tracker, but his/her Euclidean distance is greater than the threshold τ_d. Finally, we also add to the counts the

number of unwanted tracks, mostly caused by noise, given by the subtraction, at time t, of the number of valid trackers (with detections associated to them for a γ_{min} number of frames) by the number of trackers identified by the operator.

Still regarding the Equation (16), the ID_t value is the number of identity switches in a given frame at time t and c_{ID} is the weight considered for identity switches, with a value of $\log_{10}$ as proposed by the authors in [21]. In case the number of changes is equal to 0, we replace the factor $c_{ID} \cdot ID_t$ in Equation (16) by 0.

Table 3 shows our players tracking results for the *Official Match* dataset. Similarly to the previous section, we compute the F-Score measure to support our analyses. Fig. 6 shows the impact of varying the distance threshold τ_d over the MOTA and F-Score values, as well as over the global mean error obtained. We calculate this error in the same way as the previous one, but now the Euclidean distance is given between the estimated position of a true positive track and the feet position defined by the box in the ground truth.

Our results show that the particle filter-based tracking is able to track players and link their positions over time. As expected, when we increase the distance threshold τ_d, more tracks are considered as true positives, but the global error is also increased, since we take into account tracks with larger differences between the estimated position and the manually annotated position. Either way, the global mean error for players tracking is less than 35 cm for this sequence, which is again an outstanding result if we take into consideration the size of the court. In addition, it is noteworthy that the proposed methodology uses a smaller number of particles ($N = 350$) than other solutions that perform players tracking using particle filters (such as [30] that uses 500 particles), what results in processing frames faster. Consequently, the proposed system requires 25 milliseconds on average to process a frame, on a computer with an Intel Core i7 processor at 3.4 GHz, with 8 cores and 8GB of RAM.

In the experiments, we obtained successful tracking results for most part of the players. In situations where the trajectories of two athletes cross quickly, the confusion caused by the short proximity between them is resolved by the filter's prediction, based on the motion model considered.

However, there are some situations in which the filter leads to wrong estimates, mostly caused by detection problems. For example, when the players are not detected, or they are very close to each other for a significant period of time, being detected as a single blob that still meets the geometric constraints. In such cases, the filter may switch their identities, wrongly estimate their position or even the tracker can be removed, since there is no associated detection to adjust its estimate during this time. Thus, players stop being properly tracked. The detection of the ball and other objects that cause false positives can also steal the tracker of a player and spoil his/her tracking, since the observation does not distinguish those detections. This way, as futsal is a very complex contact sport and given the difficulties in adjusting the filter's estimate, the operator assistance and his/her intervention are key parts of the proposed system for achieve its goal. In all previous cases, the system operator must undo any confusion situations and manually re-identify each tracker, so they can track again the corresponding athlete.

Table 4 shows the maximum, minimum and mean durations of the tracking for all players and for the sequence itself, before a tracker is removed or has its identity switched. Players are numbered sequentially at the frame when the match starts, from left to right and from top to bottom. Thus, players from the team attacking from left to right are numbered from 1 to 5, and the ones attacking from right to left are numbered from 6 to 10. The numbers of the goalkeepers of each team in this sequence are 2 and 10, respectively. In addition, there are some substitutions of players in the match, during the recorded sequence. The substitutes who subsequently come into play are numbered according to the entry order, receiving numbers from 11 to 15 for the first team and from 16 to 20 for the second one. Thus, for example, the second substitute player to enter the game in the second team gets the number 17. As shown in Table 4, the lifespan of a tracker in this sequence is 4,635 frames in the best case, 15 frames in the worst case and 283 frames on average, what leads the operator having to make some interventions to restore the identifications of the trackers.

5.1.3 Statistical Data Computation

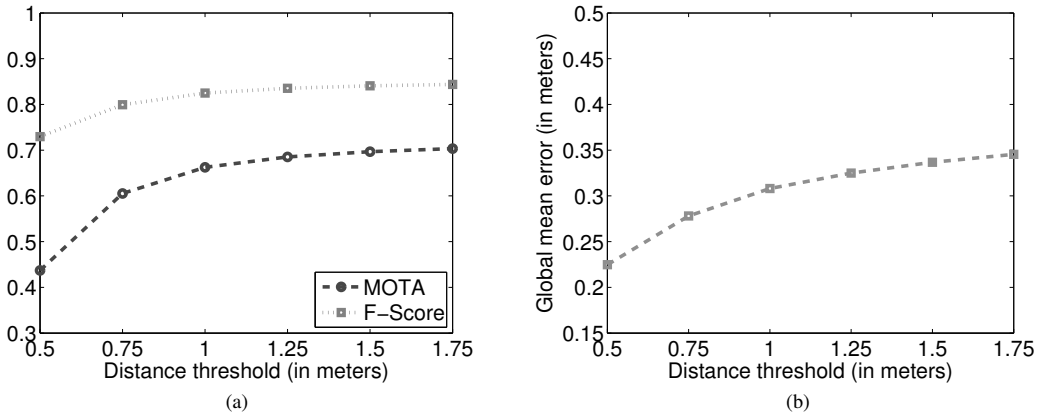
Before we present the statistical data for the *Official Match* dataset, we describe the statistical test used to define the

Table 3 Players tracking results for the *Official Match* dataset.

τ_d	TP	FP	FN	ID	Prec.	Rec.	F-Score	MOTA
0.50	5358	2268	1702	115	0.703	0.759	0.730	0.437
0.75	6210	1416	1702	115	0.814	0.785	0.799	0.605
1.00	6548	1078	1702	115	0.859	0.794	0.825	0.663
1.25	6690	936	1702	115	0.877	0.797	0.835	0.685
1.50	6765	861	1702	115	0.887	0.799	0.841	0.697
1.75	6808	818	1702	115	0.893	0.800	0.844	0.703

Table 4 Tracking duration (in number of frames) for each player.

Player	Largest duration	Shortest duration	Mean duration
1	1,085	16	225
2	1,908	15	243
3	1,044	15	243
4	1,049	15	201
5	1,030	15	245
6	2,112	15	326
7	1,691	22	463
8	1,500	17	495
9	4,635	15	456
10	499	16	88
11	309	16	135
16	1,290	54	351
17	432	25	143
18	1,089	30	343
Sequence	4,635	15	283



(a) (b)

Fig. 6 Players tracking results for the *Official Match* dataset, considering different values of distance threshold (τ_d). (a) MOTA and F-Score values as functions of the distance threshold used. (b) Global mean errors obtained according to the threshold used.

frame rate for physical data calculation. Moreover, we evaluate the error we get for physical data when we use the centroid of the bounding rectangle of the detection and when we use the feet position given by this rectangle. To determine the configuration that minimizes the errors in those calculations, we use the Analysis of Variance (ANOVA) [29]. The analysis of variance performed has two factors, with the first one having two levels (feet position or centroid position), and the second one having three levels (the possible sampling rates for computing the physical statistics, namely, every 5, 10 or 15 frames).

To obtain the samples used in ANOVA, we recorded seven extra sequences with different durations, during a training session. In those videos, we asked two athletes to travel predefined trajectories on the court, so that the distances covered by them were known. We also asked the players to move as in a real game, that is, varying times when they were running, walking or standing still on the court. Then, each player in each sequence was tracked by our system and the results for the traveled distances were computed. As the speeds are derived from the distance data, only these latter were evaluated in the test. Finally, we computed the errors for the estimated traveled distance by each player in each sequence and we performed the analysis of variance.

The result of the ANOVA has shown that, with 95% of significance, there is no statistical evidence that there is a difference between using the centroid or the feet. Thus, as the feet position can better represent the athlete's position on court in the heat map, dealing with the properties of images formed by perspective projection, we decided to use it in the system. On the other hand, the ANOVA pointed out with the same level of significance that there is, indeed, differences when we use sampling rates of 5, 10 or 15 frames. In order to define which rate introduces the smallest error, we perfomed a multiple comparison test called Tuckey's test [29]. This test has demonstrated, again with 95% significance level, that the 15 frames rate produces the smallest error, and for this reason was the one used by our system.

That said, we present in the following the statistical data computed for the *Official Match* dataset. Table 5 shows the physical data extracted for each player in this sequence. The numbering of athletes remains the same from previous section. To calculate the errors, we subtract the statistics obtained by our system from their corresponding ones in the ground truth. Finally, we divide the results of those subtractions by the statistics present in the ground truth. If an error has a negative value, the value of its corresponding statistic is less than the expected value, given the manually annotated positions on the ground truth. Similarly, an error with positive value means that its corresponding statistic exceeds the expected value for it. In addition to the individual values of each player, we also calculate the global error for the sequence, as the root mean square error (RMSE). As we can see in the Table 5, this sequence has a mean error of 8.16% for the distance traveled by the players, 8.85% for the mean speed of the athletes and 15.46% for the maximum speed. Considering the complexity of futsal and the difficulties encountered by the system in players detection and players tracking tasks, those results are very encouraging.

Regarding the errors observed, their occurrences are related to several factors. Negative errors for the distance are mainly related to players that are not detected and tracked for several frames, given the high similarity of their uniforms with the court floor, for example. This obviously reduces the estimated distance. On the other hand, the absence of a detection to adjust the filter's estimate and the problem of size changes of the bounding rectangle can increase the traveled distance for a player beyond the expected. In addition, the possible theft of a tracker by the ball or by other false positives and the identity switches also contribute to change the statistics values, until the operator can undo those confusion situations. As the calculation of players' speeds is

Table 5 Physical data estimated for each player in the *Official Match* dataset. Errors are relative to the data calculated by the ground truth positions.

Player	Dist. (m)	Dist. Error (%)	Mean Speed (Km/h)	Mean Speed Error (%)	Max. Speed (Km/h)	Max. Speed Error (%)
1	744.24	2.0	6.25	2.2	29.79	11.8
2	354.54	5.7	2.98	5.8	22.95	48.1
3	629.35	-3.8	6.42	-2.7	25.49	7.0
4	682.42	0.2	5.74	0.5	23.21	-1.9
5	641.70	1.8	5.42	2.5	29.98	20.3
6	686.39	5.8	6.35	5.8	32.80	-4.7
7	721.31	0.7	7.26	1.0	25.19	-14.6
8	663.67	3.9	7.54	4.2	25.91	-6.0
9	841.91	3.6	7.07	3.7	25.33	-1.8
10	259.87	-25.7	2.20	-25.1	16.60	8.5
11	105.73	0.1	5.72	6.1	19.10	8.8
16	223.66	8.8	7.53	11.8	19.69	1.6
17	69.67	6.6	6.35	10.7	18.37	-0.7
18	134.73	5.9	7.08	7.5	19.35	-1.6
Sequence (RMSE)	-	8.16	-	8.85	-	15.46

directly related to the computation of the players' traveled distances, those problems are also reflected in those data. The maximum speed, specifically, is mostly affected by such situations. For example, when a tracker happens to track the ball, instead of a tracking the player, obviously the maximum stored speed will be much higher than that performed by the athlete. In general, when there is an identity switch, the tracker changes its position very quickly in a short period of time, which also results in erroneous values for the maximum speed. A tracker that is lost (with no detection associated to it) and then starts to be associated with detections that are distant to it, also makes a rapid change in its position, thus generating an unreal value of maximum speed for the player. Regarding the tactical data, Fig. 7(a) shows an example of the resulting heat map for a first-team player (number 4) with mean error for physical data in this sequence. To assess the quality of the heat map, we draw black points over the image with positions given by the manually annotated coordinates in the ground truth of the player (see Fig. 7(b)).

From Fig. 7(b), we note that the heat map contains most part of the ground truth points and it is able to describe the occupancy of the player. Regions in red and yellow concentrate a large number of points very close to each other, which reveals that the player was present more often in those places. Regions where he/she was present for a short period of time are blue and contains less and more spaced points. Some ground truth points are present in regions where the player was not tracked and, similarly, the tracker recorded the passage of the player in other parts of the image, but the athlete was not really present in those locations. Some deviations between the estimated and the ground truth positions are caused by the tracking errors discussed previously.

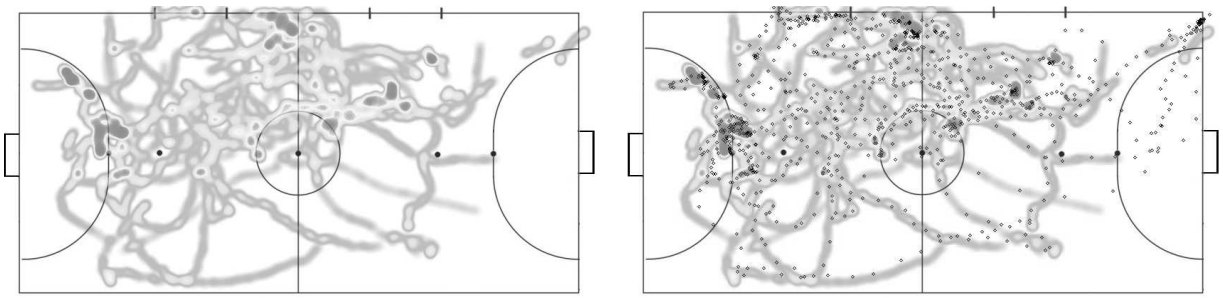
5.2 Training Match Dataset

The second dataset used consists of an image sequence captured from an training futsal match, which contains 13,320 frames and, similarly to the *Official Match* dataset, corresponds approximately to a 7 minute long video. Training matches are used to improve technical and tactical skills, to complement the physical training or even for non competitive periods during the season. In a training match, physical and tactical aspects may be detailed analyzed, especially, if the players' statistical data are provided.

Given that a training match is a more controlled scenario, we have asked the technical staff and substitute players to remain outside the court area and that substitutions were not made during the match. In case of an interruption event, such as fouls, corners or after goals scored, the game was restarted as soon as possible. The players were also asked to wear futsal training vests a little more discriminating than their usual uniforms, in order to facilitate their detection by our system. Finally, there were no referees present. Since there were no "intruders" near the court's boundaries, the restrictions with respect to the playing area were relaxed, so that a player could get out of the pitch and still be detected, up to a maximum of 2.5 meters away.

5.2.1 Players Detection Results

To evaluate the detection of players in the *Training Match* dataset, we used again the MODA CLEAR metric [21], described in Section 5.1.1. Table 6 shows the players detection results obtained. The N-MODA and F-Score measures, as well as the global mean error were calculated in the same manner as in Section 5.1.1. Fig. 8 shows the impact of varying τ_{ov} value over the N-MODA and F-Score values and over the global mean error obtained. Note that the N-MODA and F-Score values are larger for the training match than for the official match for all values of τ_{ov} considered. This was expected, since in the training match, the number of correct detections is superior, while the numbers of false positives and false negatives are smaller in almost all levels. However, note that the global mean error obtained was larger than the one computed for the official match, for all values of τ_{ov}



(a) (b)

Fig. 7 (a) Heat map of player 4 in the *Official Match* dataset. (b) Heat map of player 4 overlaid by the annotated player's positions (black points).

considered. This can be explained by the larger number of true positives obtained in the *Training Match* dataset.

The reduction in the number of false positives may be explained by the absence of referees, technical staff members and substitute players in the court area. However, the ball still was detected in some frames, as well as some players shadows that were not completely removed in the detection process. Additionally, in situations where multiple players were very close to each other and were detected as a single blob, the recursive splitting process could generate additional incorrect bounding boxes, that did not correspond exactly to the regions of the players. This has also contributed to the number of false positives obtained.

On the other hand, the training vests contributed to the their detections in some areas of the court, resulting in fewer false negatives. Finally, the relaxation of the court's dimensions has contributed to the detection of players that moved near the boundaries of the court and, thus, to the reduction in the number of false negatives as well.

5.2.2 Players Tracking Results

As performed in Section 5.1.2, we have evaluated the players tracking results in the *Training Match* dataset by using the MOTA CLEAR metric [21]. Table 7 presents the players tracking results obtained. The MOTA and F-Score measures, as well as the global mean error were calculated in the same manner as in Section 5.1.2. Fig. 9 shows the impact of varying τ_d value over the MOTA and F-Score values and over the global mean error obtained. Again, the MOTA and F-Score values are larger for the training match than for the official match for all values of τ_d considered. The improvement in the tracking results for the training match is directly related to the improvement in the players detection. Nevertheless, once again, the global mean error obtained was larger than the one computed for the official match, for all values of τ_d considered. The reason for this increase is the same one pointed out in Section 5.2.1 and is directly related to the increase in the number of true positives. Anyway, the values of errors determined are less than 40 cm, what is a promising result considering the court's dimensions.

Finally, Table 8 shows the maximum, minimum and mean durations of the tracking for all players and for the sequence itself, before a tracker is removed or has its identity switched. Players are numbered following the procedure described in Section 5.1.2. Given the players detection improvements and the consequent reduction of some problems in the tracking, it was possible to increase the maximal and mean lifespans of a tracker for this dataset. As shown in Table 8, the lifespan of a tracker in this sequence is 5,724 frames in the best case, 15 frames in the worst case and 795 frames on average.

5.2.3 Statistical Data Computation

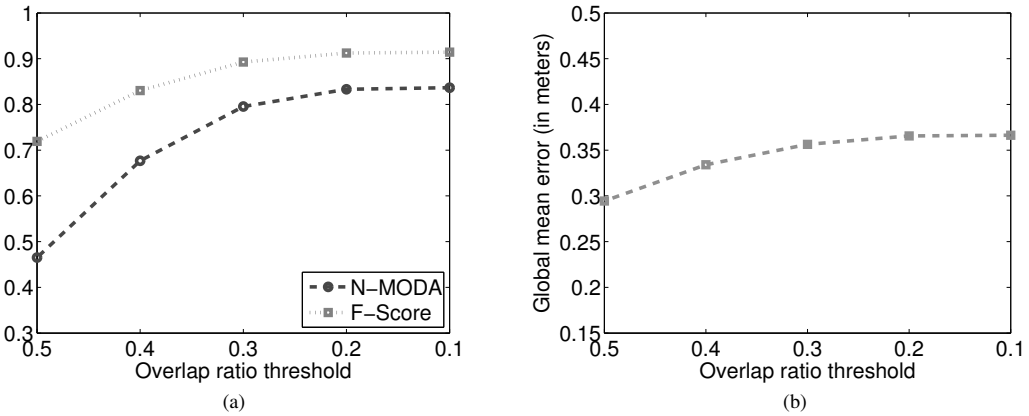
Table 9 shows the physical data extracted for each player in the *Training Match* dataset. The numbering of athletes remains the same from previous sections and the errors are calculated as described in Section 5.1.3. With the improvements in detection and tracking of athletes, we have reduced the RMSE for this dataset to 5.77% for the distance covered and 5.84% to the mean speed of the players. However, the error with respect to the maximum speed increased to 24.21%, what is justified by a higher occurrence in this dataset of events, such as, the theft of a tracker by the ball and identity switches. Note that as players are detected and tracked

Table 6 Players detection results for the *Training Match* dataset.

τ_{ov}	TP	FP	FN	Prec.	Rec.	F-Score	N-MODA
0.1	7787	308	1152	0.962	0.871	0.914	0.837
0.2	7771	324	1168	0.960	0.869	0.912	0.833
0.3	7603	492	1336	0.939	0.851	0.893	0.796
0.4	7072	1023	1867	0.874	0.791	0.830	0.677
0.5	6126	1969	2813	0.757	0.685	0.719	0.465

Table 7 Players tracking results for the *Training Match* dataset.

τ_d	TP	FP	FN	ID	Prec.	Rec.	F-Score	MOTA
0.50	5608	2784	876	71	0.668	0.865	0.754	0.435
0.75	7052	1340	876	71	0.840	0.890	0.864	0.720
1.00	7640	752	876	71	0.910	0.897	0.904	0.809
1.25	7864	528	876	71	0.937	0.900	0.918	0.839
1.50	7963	429	876	71	0.949	0.901	0.924	0.852
1.75	8007	385	876	71	0.954	0.901	0.927	0.858



(a) (b)

Fig. 8 Players detection results for the *Training Match* dataset, considering different values of overlap ratio threshold (τ_{ov}). (a) N-MODA and F-Score values as functions of the overlap ratio threshold used. (b) Global mean errors obtained according to the threshold used.

for more frames on this dataset, the majority of errors for the distance traveled are positive. The exception to this observation is the goalkeeper (number 10), who is frequently located in a blurry and noisy part of the image plane, thus making his detection and tracking harder by our system.

6 Concluding Remarks

This work presents a new system to support tactical and physical analyses of futsal teams based on computer vision. Unlike other approaches in literature, the proposed system uses a single stationary camera that captures top-view images of the court. This configuration provides some advantages. Firstly, the images obtained by a camera arranged in this manner minimize the effects of occlusion between the players, which can greatly hamper the tracking of athletes. Furthermore, the use of a single camera may reduce the computational complexity of the system and the costs for acquisition and installation.

An adaptive background subtraction technique based on Gaussian mixture is used to detect players. This technique was very effective to find the regions that correspond to the players, without an expensive training phase as performed by other approaches and without having to manually specify the objects in the scene that must be found. However, some difficulties arising from the use of background subtraction were found during the experiments. A key one is the failure to detect players when some of them are very close. In this case, the subtraction operation may provide a unique region, which results in only one valid detection containing all players. To make this problem manageable, we have applied a recursive splitting approach of that region, based on geometric characteristics of the players.

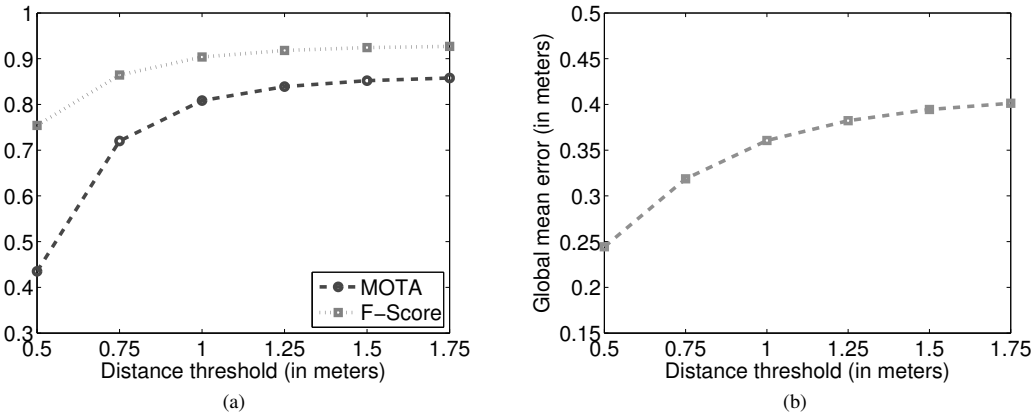
Another important reason observed for the non-detection of players was the high similarity of their appearances to some areas of the court. This problem was intensified by the characteristics of the camera used in our experiments. The images acquired had low resolution and sometimes were blurred and noisy. Additionally, the system performed some undesirable detections of the ball and shadows that had not been appropriately removed. Despite these complications, it was possible to perform the detection of athletes efficiently and accurately. The system has obtained good values for the N-MODA and F-Score measures in our experiments, as well as global mean tracking errors below 40 cm, which is a quite positive result when the court's dimensions are considered.

The particle filter method was successfully used to connect the detections of players in successive frames and make their tracking. In fact, most of the players trajectories in the datasets have been accurately estimated. The mean tracking times for the *Official Match* and *Training Match* datasets were 283 frames and 795 frames, respectively. This means that the system has demanded on some interventions of its operator to re-identify the trackers and undo situations of confusion. In most cases, tracking errors were caused by the absence of a measurement able to adjust the estimate of the filter, or by the fact that the observation model did not distinguish between the characteristics of the measurements.

Despite all the difficulties faced, our system was capable to extract the tactical and physical information of interest. The errors obtained for the players distances covered and

Table 8 Tracking duration (in number of frames) for each player.

Player	Largest duration	Shortest duration	Mean duration
1	5724	1530	3319
2	3000	90	1018
3	2142	30	730
4	1770	30	728
5	2979	30	876
6	1140	15	210
7	1956	15	304
8	1305	16	276
9	1584	17	294
10	1697	15	192
Sequence	5724	15	795



(a) (b)

Fig. 9 Players tracking results for the *Training Match* dataset, considering different values of distance threshold (τ_d). (a) MOTA and F-Score values obtained and the impact of the distance threshold on the results. (b) Global mean errors obtained according to the threshold used.

mean speeds are of the order of 8% for the *Official Match* dataset and 5% for the *Training Match* dataset, demonstrating the high potential of our system. However, the estimate of the maximum speed was more sensitive to errors, thus having errors on the order of 15% and 24% for the *Official Match* and *Training Match* datasets, respectively.

As future work, we plan to implement new methods to manage identity switches and tracking interruptions, so that fewer interventions by the system operator are necessary and our solution can be used in real time. To achieve this goal, we intend to explore alternative players detection and tracking methods. We also intend to investigate the use of alternative motion models that can better capture the dynamics of futsal players and be used in the propagation of particles in our players tracking module. Moreover, we intend to apply data mining algorithms on the collected data, to extract semantic information that is not directly obtained by simple observation. We believe that this information may reveal valuable strategies as, for example, the players positions in the court that are more likely to result in a goal. This kind of approach could be considered as an additional effort of the emerging field of sports spatiotemporal analytics [27]. Aproaches from that field will can develop robust representations from noisy or impartial data of a match, learn team behaviors in an unsupervised or semi-supervised manner and predict future behaviors. This will certainly improve decision making in different sports areas, such as, coaching, broadcasting and betting.

Acknowledgements The authors thank the support of CNPq under Processes 468042/2014-8 and 313163/2014-6, FAPEMIG under Process PPM-00542-15, CEFET-MG, CAPES and Minas Tênis Clube.

References

1. Catapult. http://www.catapultsports.com. Last access: 30/06/2015
2. Dartfish. http://www.dartfish.com/. Last access: 01 July 2015
3. Inmotio. http://www.inmotio.eu. Last access: 30 June 2015
4. Opta. http://www.optasports.com/. Last access: 01/07/2015
5. SportVU. http://www.stats.com. Last access: 30/06/2015
6. StatDNA. https://www.statdna.com/n. Last access: 01/07/2015
7. Zebra. https://www.zebra.com/. Last access: 1/07/2015
8. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Transactions on Signal Processing **50**(2), 174–188 (2002)
9. Beetz, M., Kirchlechner, B., Lames, M.: Computerized real-time analysis of football games. IEEE Pervasive Computing **4**(3), 33–39 (2005)
10. Ben Shitrit, H., Berclaz, J., Fleuret, F., Fua, P.: Multi-commodity network flow for tracking multiple people. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(8), 1614–1627 (2014)
11. Berclaz, J., Fleuret, F., Turetken, E., Fua, P.: Multiple object tracking using k-shortest paths optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(9), 1806–1819 (2011)
12. Borriello, G.: Bayesian filters for location estimation. IEEE Pervasive Computing pp. 24–33 (2003)
13. Chen, H.T., Chou, C.L., Fu, T.S., Lee, S.Y., Lin, B.S.P.: Recognizing tactic patterns in broadcast basketball video using player trajectory. Journal of Visual Communication and Image Representation **23**(6), 932–947 (2012)
14. Chen, Z.: Bayesian filtering: From kalman filters to particle filters, and beyond. Statistics **182**(1), 1–69 (2003)
15. Dearden, A., Demiris, Y., Grau, O.: Tracking football player movement from a single moving camera using particle filters. In: Conference on Visual Media Production, pp. 29–37 (2006)
16. D'Orazio, T., Leo, M.: A review of vision-based systems for soccer video analysis. Pattern recognition **43**(8), 2911–2926 (2010)
17. Figueroa, P.J., Leite, N.J., Barros, R.M.: Background recovering in outdoor image sequences: An example of soccer players segmentation. Image and Vision Computing **24**(4), 363–374 (2006)
18. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multicamera people tracking with a probabilistic occupancy map. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(2), 267–282 (2008)
19. Gedikli, S., Bandouch, J., von Hoyningen-Huene, N., Kirchlechner, B., Beetz, M.: An adaptive vision system for tracking soccer players from variable camera settings. In: International Conference on Computer Vision Systems (2007)
20. Joo, S.W., Chellappa, R.: A multiple-hypothesis approach for multiobject visual tracking. IEEE Transactions on Image Processing **16**(11), 2849–2854 (2007)

Table 9 Physical data estimated for each player in the *Training Match* dataset. Relative errors to the data calculated by the ground truth positions.

Player	Dist. (m)	Dist. Error (%)	Avg. Speed (Km/h)	Avg. Speed Error (%)	Max Speed (Km/h)	Max Speed Error (%)
1	324.85	0.3	2.64	0.4	16.32	24.2
2	801.80	6.0	6.51	6.1	27.24	3.7
3	815.33	5.3	6.60	5.2	28.96	38.5
4	889.85	7.4	7.22	7.5	25.37	-12.7
5	840.37	6.2	6.82	6.3	24.07	2.6
6	788.78	3.6	6.40	3.7	25.38	17.7
7	802.92	1.4	6.52	1.5	32.82	-2.0
8	899.27	4.5	7.32	4.9	31.81	39.6
9	743.21	7.6	6.03	7.7	22.29	7.3
10	293.03	-9.1	2.38	-9.0	19.58	40.9
Sequence (RMSE)	-	5.77	-	5.84	-	24.21

21. Kasturi, R., Goldgof, D., Soundararajan, P., Manohar, V., Garofolo, J., Bowers, R., Boonstra, M., Korzhova, V., Zhang, J.: Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(2), 319–336 (2009)
22. Khatoonabadi, S.H., Rahmati, M.: Automatic soccer players tracking in goal scenes by camera motion elimination. Image and Vision Computing **27**(4), 469–479 (2009)
23. Kim, H., Nam, S., Kim, J.: Player segmentation evaluation for trajectory estimation in soccer games. Conference on Image and Vision Computing pp. 159–162 (2003)
24. Kristan, M., Perš, J., Perše, M., Kovačič, S.: Closed-world tracking of multiple interacting targets for indoor-sports applications. Computer Vision and Image Understanding **113**(5), 598–611 (2009)
25. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
26. Liu, J., Tong, X., Li, W., Wang, T., Zhang, Y., Wang, H.: Automatic player detection, labeling and tracking in broadcast soccer video. Pattern Recognition Letters **30**(2), 103–113 (2009)
27. Lucey, P., Oliver, D., Carr, P., Roth, J., Matthews, I.: Assessing team strategy using spatiotemporal data. In: International Conf. on Knowledge Discovery and Data Mining, pp. 1366–1374 (2013)
28. Mandeljc, R., Kovačič, S., Kristan, M., Perš, J., et al.: Tracking by identification using computer vision and radio. Sensors **13**(1), 241–273 (2012)
29. Montgomery, D.C.: Design and Analysis of Experiments. John Wiley & Sons (2006)
30. Morais, E., Ferreira, A., Cunha, S.A., Barros, R.M., Rocha, A., Goldenstein, S.: A multiple camera methodology for automatic localization and tracking of futsal players. Pattern Recognition Letters **39**, 21–30 (2014)
31. Naemura, M., Fukuda, A., Mizutani, Y., Izumi, Y., Tanaka, Y., Enami, K.: Morphological segmentation of sport scenes using color information. IEEE Transactions on Broadcasting **46**(3), 181–188 (2000)
32. Nillius, P., Sullivan, J., Carlsson, S.: Multi-target tracking-linking identities using bayesian network inference. In: Conference on Computer Vision and Pattern Recognition, pp. 2187–2194 (2006)
33. Niu, Z., Gao, X., Tian, Q.: Tactic analysis based on real-world ball trajectory in soccer video. Pattern Recognition **45**(5), 1937–1947 (2012)
34. Pádua, P.H.C., Pádua, F.L.C., Sousa, M.T.D., Pereira, M.A.: Particle filter-based predictive tracking of futsal players from a single stationary camera. In: Conference on Graphics, Patterns and Images, pp. 134–141 (2015)
35. Pallavi, V., Mukherjee, J., Majumdar, A.K., Sural, S.: Graph-based multiplayer detection and tracking in broadcast soccer videos. IEEE Transactions on Multimedia **10**(5), 794–805 (2008)
36. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: European Conference on Computer Vision, pp. 661–675 (2002)
37. Perl, J., Grunz, A., Memmert, D.: Tactics analysis in soccer – an advanced approach. International Journal of Computer Science in Sport **12**(1) (2013)
38. Renno, J.P., Orwell, J., Thirde, D., Jones, G.A.: Shadow classification and evaluation for soccer player detection. In: British Machine Vision Conference, pp. 1–10 (2004)
39. Santiago, C.B., Sousa, A., Estriga, M.L., Reis, L.P., Lames, M.: Survey on team tracking techniques applied to sports. In: International Conference on Autonomous and Intelligent Systems, pp. 1–6 (2010)
40. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: Conference on Computer Vision and Pattern Recognition, vol. 2 (1999)
41. Sullivan, J., Carlsson, S.: Tracking and labelling of interacting multiple targets. In: European Conference on Computer Vision, pp. 619–632 (2006)
42. Teutsch, M.: Moving Object Detection and Segmentation for Remote Aerial Video Surveillance, vol. 18. KIT SP (2015)
43. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Conference on Computer Vision and Pattern Recognition, pp. I–511 (2001)
44. Wang, X., Ablavsky, V., Ben Shitrit, H., Fua, P.: Take your eyes off the ball: Improving ball-tracking by focusing on team play. Computer Vision and Image Understanding **119**, 102–115 (2014)
45. Wisbey, B., Montgomery, P.G., Pyne, D.B., Rattray, B.: Quantifying movement demands of afl football using gps tracking. Journal of Science and Medicine in Sport **13**(5), 531–536 (2010)
46. Xu, M., Orwell, J., Lowey, L., Thirde, D.: Architecture and algorithms for tracking football players with multiple cameras. Vision, Image and Signal Processing **152**(2), 232–241 (2005)
47. Yao, J., Odobez, J.M.: Multi-camera multi-person 3d space tracking with mcmc in surveillance scenarios. In: European Conference on Computer Vision - Workshop on Multi Camera and Multimodal Sensor Fusion Algorithms and Applications (2008)
48. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(11), 1330–1334 (2000)
49. Zhu, G., Huang, Q., Xu, C., Rui, Y., Jiang, S., Gao, W., Yao, H.: Trajectory based event tactics analysis in broadcast sports video. In: International Conference on Multimedia, pp. 58–67 (2007)
50. Zivkovic, Z.: Improved adaptive gaussian mixture model for background subtraction. In: International Conference on Pattern Recognition, pp. 28–31 (2004)