Particle Filter-based Predictive Tracking of Futsal Players From a Single Stationary Camera

Pedro H. C. de Pádua, Flávio L. C. Pádua, Marco T. D. Sousa
Piim-Lab - Department of Computing
Centro Federal de Educação Tecnológica de Minas Gerais
Belo Horizonte, MG, Brazil
{pedhenrique,cardeal,mtdsousa}@decom.cefetmg.br

Marconi de A. Pereira
DTECH
Universidade Federal de São João del-Rei
Ouro Branco, MG, Brazil
marconi@ufsj.edu.br

***Abstract*—In this paper we study the use of computer vision techniques for visual tracking of futsal players. In the sports field, player tracking is an important task, as it can provide an estimate of the position of the athlete in a given time and thus compute his/her trajectories. This information can be used by coaches and sport professionals on tactical and physical analyses. We use adaptive background subtraction and blob analysis to detect players, as well as particle filters to predict their positions and track them using data from a single stationary camera. Experimental results show that our approach is capable to track players and compute their trajectories over time with errors below 20 cm, thus demonstrating a high potential to be used in a wide range of futsal match analyses.**

***Keywords*-Player Tracking; Particle Filter; Futsal; Tactical and physical analyses.**

I. Introduction

The interest on sport analyses has growth large over the past years. In futsal, as in other team sports, tactical and physical analyses are fundamental to understand what is happening in the game, to identify and correct errors and to plain improvements. Through those observations, sportsmen can verify players physical efficiency, refine strategies and better adapt training routines [1]. To make such analyses possible, it is necessary to estimate players positions at a given instant of time and, consequently, track them [2]. From that, it is possible to compute players trajectories over time, as seen in Fig. 1, which are the core data of such analyses [3].

Unfortunately, a major part of this work is performed manually by staff members or specialized companies [4], [3], [2]. The matches are recorded in video and reviewed exhaustively so that observations are made, registered and passed later to coaches. That makes the player position estimate a time-consuming task, which is prone to human error and can demand on significant financial costs.

Even though the development of some technological solutions in recent years has helped sportsmen to track players in a automatic [5] or semi-automatic way [6], their adoption by futsal teams is still difficult, by their complexity and monetary costs. In this context, we propose a vision-based approach to automatically detect and track multiple futsal players with minimum human intervention, in such a way their positions in a given time can be estimated and their trajectories computed.

A. Related work

Many efforts have been applied to detect and track players using computer vision techniques in several kinds of sports. The images sources used by researchers differ from fixed cameras images, as in [7], [8], [4], [2], [9], [10] (which are capable, in most cases, to monitor all game actions) to moving cameras or broadcast images, in [11], [12], [13], [14], [15], [16], [17] (which are most likely easier to obtain). Each of the images sources used presents different challenges to overcome.

For player detection, some papers use techniques based on segmentation and morphological operations. A commonly adopted approach is to build a model of the player or the background based on colour information, using the predominant game region colour [7], [13] or histograms and colour distributions of players [18], that allow the extraction of regions that contain the athletes. The detection step using colour-based techniques is usually fast, despite the fact they are very sensitive to illumination variations which may reduce they precision and robustness. Adaptive background subtraction methods based on mixture of gaussians, in turn, are more resistant to illumination variations [19]. On the other hand, they are slower than colour-based methods, and if the target stays static for sufficient time, it can be incorporated by the background model and, consequently, not be detected.

Another widely used technique to locate players in images is the use of trained detectors (e.g. Cascade classifiers or Deformable Part Models (DPM)). The authors in [2], [11]

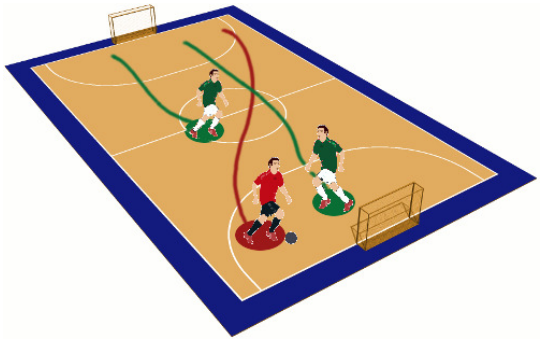


Fig. 1. Trajectories of three players highlighted in red and green: with trajectory data one can derive information about speed, distance travelled, occupancy heatmaps of the athletes, among others useful for the analyses.

use manually extracted samples from game scenes to train a haar classifier, which is used to detect players instances in images effectively. However, the training phase has, in most cases, a high complexity cost and demands on a big number of samples. Furthermore, the detection process with this technique is usually slow, which results in poor processing frame rates.

To increase the robustness of detection, some works make use of probabilistic approaches together with some of the aforementioned techniques in multi-camera or single-camera setups. Using multiple cameras, the authors in [2] combine the detections made by haar detectors in images from each camera in a multiple-hypothesis function, that represents the likelihood of a player be found in a certain court position. This function is built through the projection, in a virtual calibrated court plane, of the player location image coordinates. Similarly, the authors in [9], [10] use background subtraction together with the Probabilistic Occupancy Map (POM) technique to detect the players in different situations. Considering a single-camera setup, the authors in [12] make use of likelihood maps based on colour distributions of players to estimate their locations.

At the same time, different approaches can be found in the literature regarding multiple players visual tracking. One widely adopted technique is the use of predictive filters, as Kalman Filter or Particle Filter. To estimate the speed and position of players and link their trajectories, the authors in [20] use Kalman Filter. However, Kalman Filter is not adequate for multiple-hypothesis processes and, for this reason, numerous authors choose to use Particle Filter to track players and model their motion [11], [13], [14], [15], [4], [2]. In such cases, the complexity cost increases proportionally with the numbers of players tracked.

There are also works that explore trajectory analyses and graph-based multiple-hypothesis to perform player tracking. In that case, graphs that represent the possible player trajectories are built, modelling their position in a given instant of time along with their transitions between frames [8], [7], [16], [9], [10]. The trajectories of players are then searched in the graph using a similarity measure [8], linear programming [9], multi-commodity network flow [10] or modelled as a minimum edge cover problem [16]. Commonly, graph-based methods have a high complexity cost, so it is difficult to achieve good processing frame rates using this type of technique to be used in real time applications.

B. Contribution

In this paper, we explore some of the aforementioned techniques to detect and track players allying high processing frame rates with acceptable robustness levels. Fig. 2 presents an overview of the proposed approach and the necessary steps to compute players trajectories. Unlike most part of the previous works, we use a single stationary camera, c, placed so its optical axis, o, is approximately perpendicular to the ground. In such configuration, the camera can monitor the entire court area and capture top-view images of the court, consequently minimizing the undesired effects of occlusions among players. To capture the entire court, we make use of wide-angle lens, but they undesirably causes substantial spherical distortion on the images. We estimate camera parameters using the algorithm proposed in [21] and undistort the images to increase the precision of our approach.

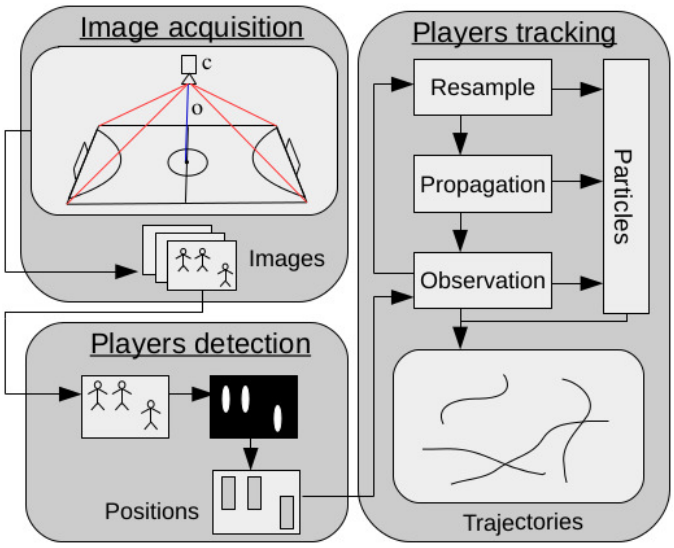


Fig. 2. Overview of our approach and the steps necessary to calculate players trajectories.

In the players detection step, we use an adaptive background subtraction method based on a mixture of gaussians [19] and on geometric constraints to check blobs sizes in the resulted binary image. To track players in successive frames and estimate their positions at a given time, we use particle filters. Each player detected is automatically tracked by a separate particle filter. The data association between detections and trackers in each frame, present in multi-target tracking problems, is solved using the well-known Hungarian Algorithm [22]. If a detection is associated to a tracker, it is used to guide the particles of the associated tracker. Otherwise, filters prediction is used to estimate the position of the player at that time.

That said, the main contribution of this work consists in to present a simple yet efficient low cost approach to track futsal players, as shown in our experimental results, that is capable of providing trajectories of players to be used in a wide range of analyses. The remainder of this paper is organized as follows. Section II presents the details of our player detection approach. Section III describes the tracking methodology used in this paper. Experimental results and discussions are presented in Section IV, followed by the conclusions and suggestions for future work in Section V.

II. Players Detection

In our methodology, the first step for detecting players consists in to perform a background subtraction to segment them from the game region. In general, the background presents some regular behaviour 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. The background

subtraction result is an image that highlights the regions of non-stationary objects in the scene.

In this paper, we make use of an adaptive background subtraction technique, based on Gaussian Mixture Models (GMM), as proposed in [19]. In this technique, the background (BG) model is estimated from a training set denoted as χ. This training set is built from pixels values $\vec{x}$ sampled over a time adaptation period T, so that at time t we have $\chi_T = \{\vec{x}^{(t)}, \vec{x}^{(t-1)}, ..., \vec{x}^{(t-T)}\}$, and the estimated background model is denoted by $\hat{p}(\vec{x}|\chi, BG)$ [19].

Each new sample is incorporated to the set and the old ones are discarded, so the model is updated in order to adapt to changes. In the recent samples, however, there could be some values that belong to background as well to foreground (FG) objects. This estimative should be denoted by $p(\vec{x}^{(t)}|\chi_T, BG+FG)$ and in a GMM with M components, is given by [19]:

$$\hat{p}(\vec{x}|\chi_T, BG+FG) = \sum_{m=1}^{M} \hat{\pi}_m \mathcal{N}(\vec{x}; \hat{\vec{\mu}}, \hat{\sigma}_m^2 I), \quad (1)$$

where $\hat{\vec{\mu}}_m$ is the estimate of the mean of the m^{th} Gaussian component and $\hat{\sigma}_m$ is the estimate of the variances that the describe the m^{th} Gaussian component. The covariance matrices are assumed to be diagonal and I, the identity matrix, has proper dimensions [19]. The mixing weights (the portion of data accounted by the m^{th} Gaussian), denoted by $\hat{\pi}_m$, are non-negative and normalized so they sum to one.

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 be part of the background. 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 $\hat{\pi}_m$ [19]. The background model can be approximated by the first B largest clusters:

$$p(\vec{x}|\chi_T, BG) \sim \sum_{m=1}^{B} \hat{\pi}_m \mathcal{N}(\vec{x}; \hat{\vec{\mu}}_m, {\sigma^2}_m I). \quad (2)$$

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

$$B = \arg\min_b \left(\sum_{m=1}^{b} \hat{\pi}_m > (1 - c_f) \right), \quad (3)$$

where c_f is a measure of the maximum portion of data that can belong to foreground objects without influencing the background model [19]. This way, the first B of the ranked components whose weights exceed $(1 - cf)$ 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 [19] 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 $\vec{x}^{(t)}$ at time t, the recursive update equations are:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha(o_m^{(t)} - \hat{\pi}_m) - \alpha c_T \quad (4)$$

$$\hat{\vec{\mu}}_m \leftarrow \hat{\vec{\mu}}_m + o_m^{(t)}(\alpha/\hat{\pi}_m)\vec{\delta}_m \quad (5)$$

$$\hat{\sigma}_m^2 \leftarrow \hat{\sigma}_m^2 + o_m^{(t)}(\alpha/\hat{\pi}_m)(\vec{\delta}_m^T \vec{\delta}_m - \hat{\sigma}_m^2), \quad (6)$$

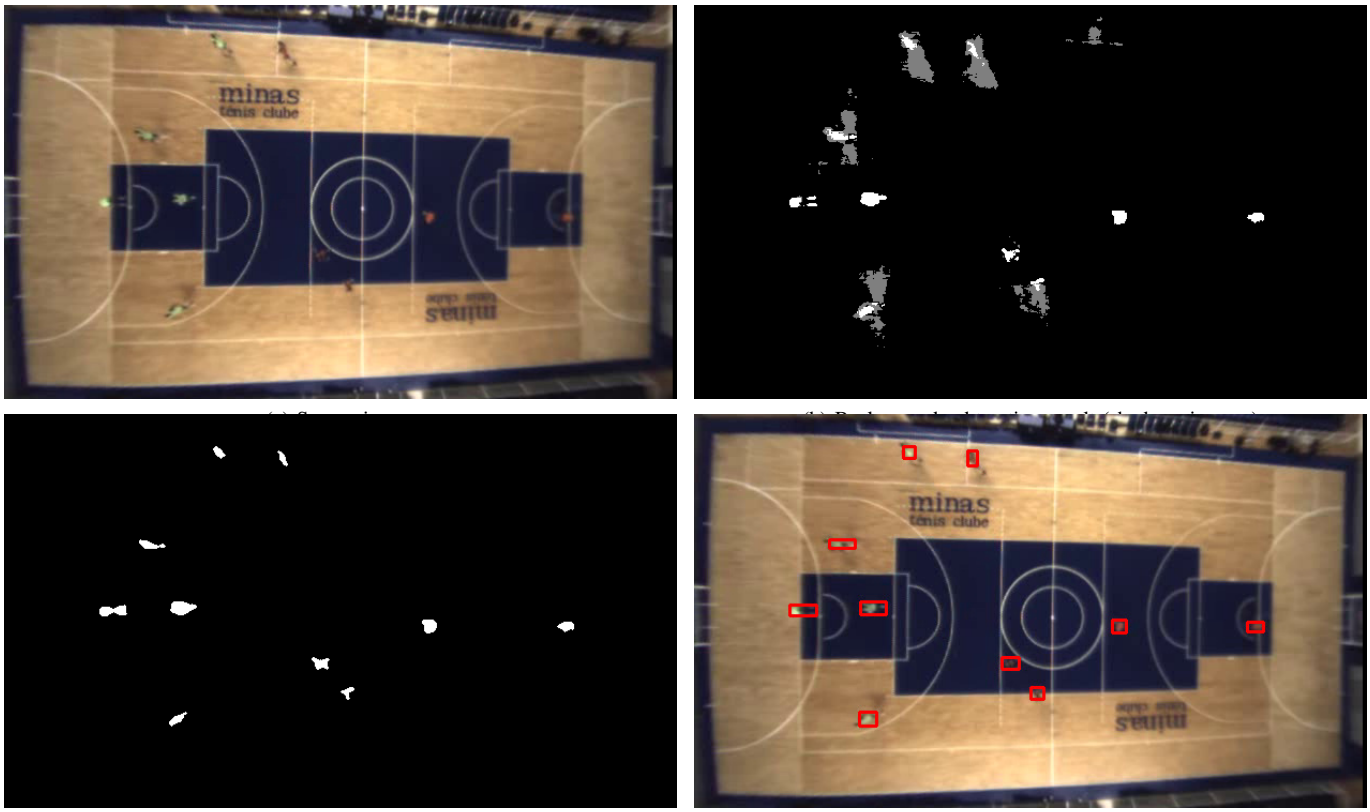
where $\vec{\delta}_m = \vec{x}^{(t)} - \hat{\vec{\mu}}_m$. The constant α describes an exponentially decaying envelope, used to limit the influence of the old samples and, approximately, $\alpha = 1/T$. For a new sample, the ownership $o_m^{(t)}$ is set to 1 for the "close" component with the largest weight $\hat{\pi}_m$ 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 m^{th} component is calculated as $D_m^2(\vec{x}^{(t)}) = \vec{\delta}_m^T \vec{\delta}_m / \hat{\sigma}_m^2$. If there are no "close" components, a new component is generated with $\hat{\pi}_{M+1} = \alpha$, $\hat{\vec{\mu}}_{M+1} = \vec{x}^{(t)}$ and $\hat{\sigma}_{M+1} = \sigma_0$, where σ_0 is some initial variance with appropriate value [19]. If the maximum number of components is reached, the component with the smallest weight is discarded. Finally, c_T 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 centred on the first sample. New components are added or discarded as aforementioned, so the number of components is dynamically updated and the background model is effectively estimated. 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.

After the background subtraction, we get an image as shown in Fig. 3b. To increase robustness, it is necessary to detect moving shadows pixels upon pixels labelled 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 where 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 from the foreground) and closing (to remove small holes on foreground blobs). The result of this operations can be seen in Fig. 3c. At this moment, bounding rectangles are also assigned to each blob as possible players locations creating a set R of regions of interest.

Lastly, 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



(c) Morphological operations result

(d) Detection result (bounding rectangles in red)

Fig. 3. Player detection process.

positions. The i-th region in R is discarded if $w_i < w_{min}$ or $h_i < h_{min}$, where 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 $h_i > h_{max}$, where 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 consider only detections that are inside the court area, we use a homography matrix H previously estimated to make a perspective transformation between two planes, that is, transform points in image coordinates to court coordinates. To estimate H, we choose a set of known points in the court (e.g. the centre of the court, its corners, the penalty mark, among others) whose positions is given by a chosen referential and find their corresponding image points positions in pixels. With those matches, we can estimate a matrix that can maps points in the image plane to points in the court plane. Then, we use H to transform the centroid of the i-th region in R and check if its position in the court plane is inside the limits of the court. If it is not, the detection is discarded.

Fig. 3d shows detected players on the source image as a result of the detection step.

III. Players Tracking

After the detection process, the next step in our approach is to track players and estimate their positions at a given time, linking their detections over successive frames. Player tracking can be seen as an iterative process, which analyzes the image sequence to describe players motion. In this work, we make use of Particle Filter for this task.

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 [23]. 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 in the future. Whether 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 in a time t a state $X_i^{(t)}$, which contains an information that represents the player. Each particle has also

a weight ${W_i}^{(t)}$, which account how good that sample 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 ${X_i}^{(t)} = \{x, y, v_x, v_y\}$, where 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 mean state of the player tracked, $\hat{X}^{(t)}$, is given by:

$$\hat{X}^{(t)} = \sum_{i=1}^{N} {X_i}^{(t)} {W_i}^{(t)}. \tag{7}$$

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 and a position 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 generated with states ${X_i}^{(t_0)} = \{x, y, v_x, v_y\}$. The values of x and y are computed according to a normal distribution around the centre of the rectangle of the associated detection with variance ${\sigma^2}_{(x,y)}$. On the other hand, v_x and v_y are initialized with values equal to zero. All particles have the same weight, that is ${W_i}^{(t_0)} = 1/N$.

From this moment, an iterative process begins, which is repeated for every new frame in the images sequence, consisting of the *Resample*, *Propagation* and *Observation* phases. Next, we describe each one of those phases, considering t as the current time.

A. 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 greater 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 greater weights, but we still admit repetitions and also allow small weight particles to be selected.

B. Propagation

In this phase, we propagate the particle set by using the motion model to build the estimate of state $X^{(t+1)}$. 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 [2], motivated by the fact that the variations between frames are very small when images are captured at 30 frames per second. In this model, we have:

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

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

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

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

$$X^{(t+1)} = \begin{bmatrix} 1 & 0 & 1/30 & 0 \\ 0 & 1 & 0 & 1/30 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}^{(t)} + \begin{bmatrix} 0 \\ 0 \\ e_{v_x} \\ e_{v_y} \end{bmatrix} \right) \tag{10}$$

where e_{v_x} and e_{v_y} are the process noise.

We set the initial variance ${\sigma^2}_{(x,y)}$ for the position based on the average size of the player in images. During tracking, that variance decreases 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.

C. Observation

In this phase, the estimates are adjusted by an observation model 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 $P(X_{t+1}|Z_{t+1})$. As we are tracking players and estimating their positions, we adopt a model where ${Z^{(t+1)} = [x \ y]}^{\top(t+1)}$, so only the position information is considered.

However, as we deal with multi-player tracking, it is necessary, at first, to decide which detection in an image 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 applied the well-known Hungarian Algorithm [22]. This technique calculates the cost of all association possibilities, given in our work by the Euclidean Distance between a position of one detection (the centroid of the rectangle) and the position of a tracker. The algorithm makes the appropriate associations in such a way each tracker is associated with one detection at most with the smallest possible cost, in polynomial time.

With all associations made, we check if the each cost given is smaller then a threshold λ, which controls the maximum acceptable cost. We do this to minimize unreal situations, most likely cause by false positive detections (e.g. a player that is detected at the penalty mark in one frame and in

TABLE I
PARAMETERS VALUES USED IN THE EXPERIMENTS.

Parameters	Value
w_{min}, h_{min}	5 pixels
w_{max}	50 pixels
h_{max}	40 pixels
N	250 particles
$\gamma_{min}, \gamma_{lim}$	5 frames
$\sigma^2{}_{(x,y)}$	5 pixels
$\sigma^2{}_{(v_x,v_y)}$	8 pixels
θ_l	3 pixels
θ_h	7 pixels
λ	40 pixels

the next he/she is detected at the centre 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 with a tracker, this tracker uses only its prediction data and receives a strike. When a maximum number of strikes is reached, 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, we again use the Euclidean Distance d between their (x, y) positions at state $X^{(t+1)}$ and: (i) the rectangle centroid of the associated detection or (ii) the estimated (x, y) position of the player in the previously time ($\hat{X}^{(t)}$), if there is no associated detection. We use d in a normal density function given by:

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(-\frac{d^2}{2\sigma^2}\right)}, \quad (11)$$

where $\sigma = \sigma_{(x,y)}$. Finally, we normalize the weights so they sum up to one. From that, the mean state of the player can be calculated again by Eq. 7 and the filter is ready for another iteration. We store in the tracker history the estimated location of the player, given by its mean state, and we can then compute his/her trajectories over time.

IV. EXPERIMENTAL RESULTS

To demonstrate the effectiveness of our approach, we tested it on a challenging real-game sequence, collected from a training session of *Minas Tênis Clube* futsal team, one of the major teams in this sport in Brazil. This sequence consists of 13320 frames captured at 30 frames per second and on a 752x480 pixels resolution. We manually annotate the positions of the athletes with a bounding box around each player once every 15 frames, to create our ground truth for the detection and tracking experiments. The average player box in images is 20x15 pixels. For computational reasons, we crop the court region in image, resulting in 628x368 pixels frame.

As showed in the previous sections, different parameters need to be set in our approach. The parameters values are selected empirically and the values used in our experiments for the main variables are summarized in Table I.

A. Experiments for players detection

To evaluate the detection of players, we use the CLEAR metric Multiple Object Detection Accuracy (MODA) [24], which have become one of the standards for evaluation of object detection algorithms in the computer vision area. This metric utilize 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 detection region R_i found by our algorithm. For a given overlap threshold τ_{ov}, a detection D is a true positive if [25]:

$$\frac{|D_i \cap G_i|}{|D_i \cup G_i|} \geq \tau_{ov}, \quad (12)$$

where D_i and G_i is the i-th mapped pair of detection and ground truth. The choice for the value τ_{ov} may vary with the detection context. For larger objects in a image, that cover several thousands of pixels, values as 0.5 or 0.7 are suitable for the threshold. However, for small objects as in the present case, where players have an average size of 20x15 pixels, even small deviations in size or position of the bounding box annotated can induce a significant lower overlap [25]. These deviations can be caused, in our context, mostly by merged detections of two or more players or partial detections of players. In order to demonstrate the impact of the overlap threshold on the performance evaluation, we vary τ_{ov} between 0.1 and 0.5.

TABLE II
PLAYERS DETECTION RESULTS FOR THE TESTED SEQUENCE

τ_{ov}	TP	FP	FN	Prec.	Rec.	F-Score	N-MODA
0.1	7333	281	1603	0.963	0.821	0.886	0.789
0.2	7315	299	1621	0.961	0.819	0.884	0.785
0.3	7135	479	1801	0.937	0.798	0.862	0.744
0.4	6504	1110	2432	0.854	0.728	0.786	0.603
0.5	5454	2160	3482	0.716	0.610	0.659	0.368

As MODA is originally defined for single frames, we can compute the Normalized MODA (N-MODA) for the entire sequence as [24]:

$$N\text{-}MODA = 1 - \frac{\sum_{t=1}^{N_{frames}} (c_m(m_t) + c_f(fp_t))}{\sum_{t=1}^{N_{frames}} N_G^{(t)}}, \quad (13)$$

where m_t is the number of misses (false negatives), fp_t is the number of false positives and $N_G^{(t)}$ is the number of ground truth objects (TP + FN), all three for a given frame t. The weights c_m and c_f are the cost functions for the missed detections and for the false positives, respectively. Similarly to [24], in our evaluation c_m and c_f are both equal to one.

Table II shows our players detection results. For clarification purposes, we also compute the F-score for our experiments, given by the harmonic mean of precision (*Prec.*) and recall (*Rec.*) values. Fig. 4 shows the impact of the variation in the overlap ratio threshold over the N-MODA and F-score values. As aforementioned, in our context smaller overlap ratio thresholds are more appropriated and lead to significant values. As we deal with small objects in the scene, the global mean error for the detection in our sequence is below 0.20m, despite the τ_{ov} used. This is a promising value, considering the court dimensions (38x19m).

Regarding the false positives count, most part are caused by the detection of the ball or detection of light shadows that were

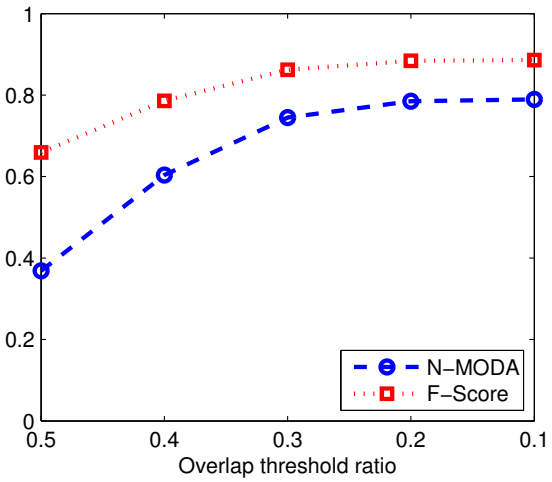


Fig. 4. Evaluation results using N-MODA and F-Score metrics and the impact of the overlap ratio threshold over the performance.

TABLE III
PLAYERS TRACKING RESULTS FOR THE TESTED SEQUENCE

TP	FP	FN	ID	Prec.	Rec.	F-Score	MOTA
2601	285	480	30	0.901	0.844	0.871	0.752

not filtered in the process. On the other hand, a significant part of the false negatives are caused by a high similarity between the background and the appearance of the player or caused by the clutter of the background. Nevertheless, the results obtained are very encouraging and demonstrate the potential of our detection approach to efficiently localize players in the images.

B. Experiments for players tracking

Similar to the previous section, we use the CLEAR metric Multiple Object Tracking Accuracy (MOTA) [24] to evaluate the tracking of the players in our sequence. To do so, again we need to compute the number of false negatives, false positives, true positives and also the number of identity switches, for a given reference ground truth track. We consider that a track is a true positive if the distance between the estimate position given by the tracker and the centroid of the ground truth box is lower than 1 meter. This way, MOTA is defined as [24]:

$$MOTA = 1 - \frac{\sum_{t=1}^{N_{frames}} (c_m(m_t) + c_f(fp_t) + c_s(ID))}{\sum_{t=1}^{N_{frames}} N_G^{(t)}}, \quad (14)$$

where c_m, m_t, c_f, fp_t and $N_G^{(t)}$ are defined in the same way as in Eq. 13, and $c_m = c_f = 1$. The value ID is the number of identity switches in the sequence and c_s is the weight function for the identity switches, equal to $\log_{10}$ as proposed by the authors in [24]. For the tracking experiments, we consider approximately 40% of our dataset, since it is very time consuming to check such a large quantity of data. This results in a 3 minute sequence, with players positions annotated once every 15 frames.

We present the tracking results in Table III. As it can be seen, the results again show that the particle filter approach is capable of tracking players efficiently and link their positions over time. However, there are some situations when the filter leads to wrong estimates. Most of them are caused by players that are not correctly detected or players that are very close to each other, being detected as a unique blob that still meets the size constraints, for a significant period of time. In such cases, the filter might switch their identities, wrongly estimate their positions or even be removed, as it does not have an associate detection in this time. The detection of the ball may also steal the tracker of a player.

During the experiments, we obtained several successful tracking for most part of players, as can be seen in Fig. 5, with global mean error once more below 0.20m for the sequence. The mean duration of the tracking, before a tracker is lost or has its identity switched, averaged for all players, is 790 frames. The lifespan of a tracker is 5724 frames in the best case and 5 frames in the worst one. Besides that, our methodology uses a lower number of particles (N = 250) than other works (e.g. 5000 in [11] and 500 in [2]), which results in faster process rates. This approach needs 16 milliseconds on average to process a frame, being able to be potentially used in a wide range of futsal match analyses. Comparatively to other works, as in [11], [15], our methodology is capable of dealing with a challenger sequence, with less visual cues to explore, without needing an expensive training phase [11] and without manual model initialization [15].

V. CONCLUDING REMARKS

In this paper, we use a single stationary camera approach to track futsal players using particle filters. Our experimental results suggest that our methodology can be successfully applied for performing accurate detection and tracking of players in real game sequences. Our approach is also able to estimate player trajectories over time, which can be further used to support different kinds of physical and tactical analyses. This methodology can also be used in other scenarios and track players in different indoor sports as hockey [11], handball or basketball [4]. The experimental results demonstrate that the proposed approach can operate with hight accuracy levels and processing frame rates, with global mean errors below 0.20m.

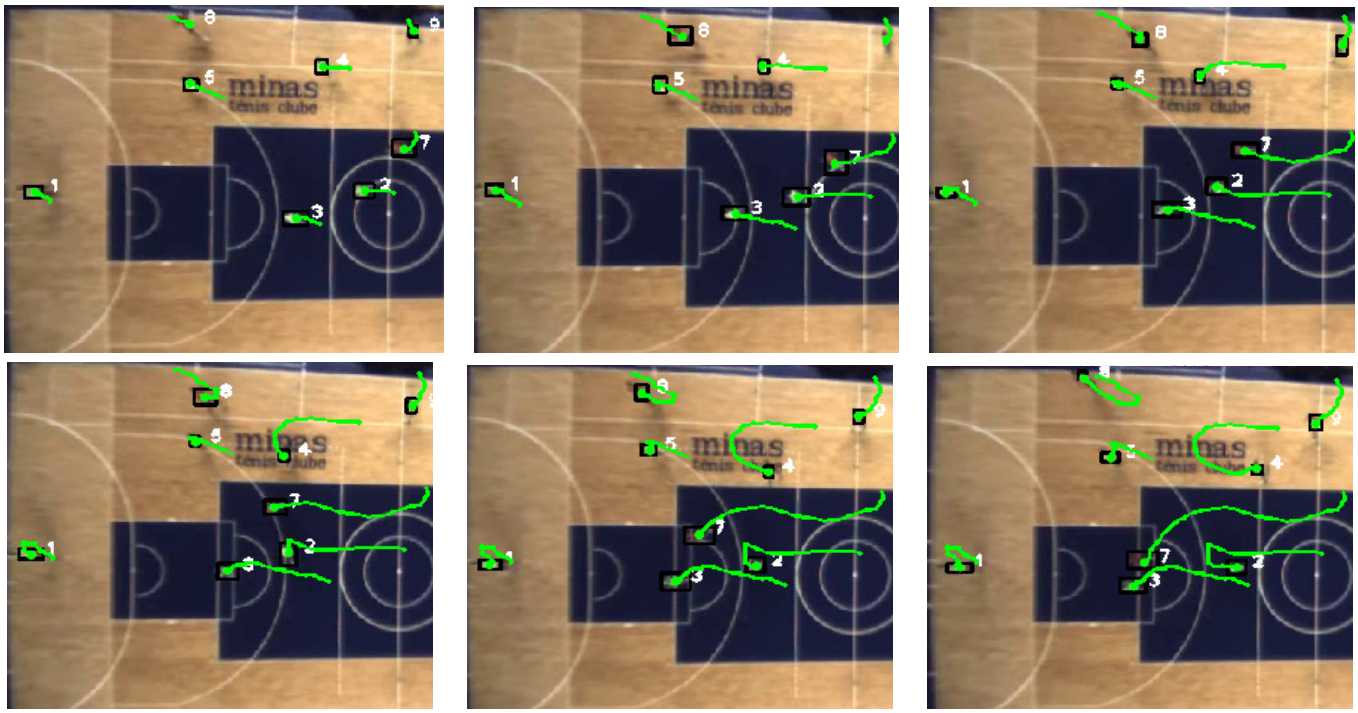
In future work, we will focus on improving our observation model to prevent confusion situations, with the use of appearance data of players in the assignment step and particle weight calculation, so minimizing identity switches. We also plan to extend the detection algorithm, making use of templates or trained classifiers to better detect players that are close to each other or that are very similar to the background.

ACKNOWLEDGMENT

The authors thank the support of CNPq-Brazil under Procs. 468042/2014-8 and 313163/2014-6, FAPEMIG-Brazil under Proc. APQ-01180-10, CEFET-MG under Proc. PROPESQ-023-076/09, of CAPES-Brazil and Minas Tênis Clube.

REFERENCES

[1] Z. Niu, X. Gao, and Q. Tian, "Tactic analysis based on real-world ball trajectory in soccer video," *Pattern Recognition*, vol. 45, no. 5, pp. 1937–1947, 2012.



(d) Frame 5430 (e) Frame 5445 (f) Frame 5460

Fig. 5. Example of successful tracking of players and their trajectories (highlighted in green) over time.

[2] E. Morais, A. Ferreira, S. A. Cunha, R. M. Barros, A. Rocha, and S. Goldenstein, "A multiple camera methodology for automatic localization and tracking of futsal players," *Pattern Recognition Letters*, vol. 39, pp. 21–30, 2014.

[3] T. D'Orazio and M. Leo, "A review of vision-based systems for soccer video analysis," *Pattern Recognition*, vol. 43, no. 8, pp. 2911–2926, 2010.

[4] M. Kristan, J. Perš, M. Perše, and S. Kovačič, "Closed-world tracking of multiple interacting targets for indoor-sports applications," *Computer Vision and Image Understanding*, vol. 113, no. 5, pp. 598–611, 2009.

[5] Stats, "SportVU," http://www.stats.com/sportvu/sportvu.asp, [Online; Accessed 26 Mar. 2015].

[6] Prozone Sports, "Prozone," http://www.prozonesports.com/, [Online; Accessed 26 Mar. 2015].

[7] P. J. Figueroa, N. J. Leite, and R. M. Barros, "Tracking soccer players aiming their kinematical motion analysis," *Computer Vision and Image Understanding*, vol. 101, no. 2, pp. 122–135, 2006.

[8] P. Nillius, J. Sullivan, and S. Carlsson, "Multi-target tracking-linking identities using bayesian network inference," in *Computer Vision and Pattern Recognition, IEEE Conference on*, vol. 2. IEEE, 2006, pp. 2187–2194.

[9] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Tracking multiple people under global appearance constraints," in *Computer Vision, IEEE International Conference on*. IEEE, 2011, pp. 137–144.

[10] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua, "Multi-commodity network flow for tracking multiple people," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 8, pp. 1614–1627, 2014.

[11] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *European Conference on Computer Vision*. Springer, 2004, pp. 28–39.

[12] M. Beetz, N. v. Hoyningen-Huene, J. Bandouch, B. Kirchlechner, S. Gedikli, and A. Maldonado, "Camera-based observation of football games for analyzing multi-agent activities," in *Proc. of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 2006, pp. 42–49.

[13] A. Dearden, Y. Demiris, and O. Grau, "Tracking football player movement from a single moving camera using particle filters," in *Proc. of the 3rd European Conference on Visual Media Production*, 2006, pp. 29–37.

[14] G. Zhu, Q. Huang, C. Xu, Y. Rui, S. Jiang, W. Gao, and H. Yao, "Trajectory based event tactics analysis in broadcast sports video," in *Multimedia, 2007. Proc. of the 15th International Conference on*, 2007, pp. 58–67.

[15] J. Czyz, B. Ristic, and B. Macq, "A particle filter for joint detection and tracking of color objects," *Image and Vision Computing*, vol. 25, no. 8, pp. 1271–1281, 2007.

[16] V. Pallavi, J. Mukherjee, A. K. Majumdar, and S. Sural, "Graph-based multiplayer detection and tracking in broadcast soccer videos," *Multimedia, IEEE Transactions on*, vol. 10, no. 5, pp. 794–805, 2008.

[17] R. Hess and A. Fern, "Discriminatively trained particle filters for complex multi-object tracking," in *Computer Vision and Pattern Recognition, IEEE Conference on*, June 2009, pp. 240–247.

[18] N. Vandenbroucke, L. Macaire, and J.-G. Postaire, "Color image segmentation by pixel classification in an adapted hybrid color space. application to soccer image analysis," *Computer Vision and Image Understanding*, vol. 90, no. 2, pp. 190–216, 2003.

[19] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, Proc. of the 17th International Conference on*, vol. 2. IEEE, 2004, pp. 28–31.

[20] M. Xu, J. Orwell, L. Lowey, and D. Thirde, "Architecture and algorithms for tracking football players with multiple cameras," *IEEE Proc. on Vision, Image and Signal Processing*, vol. 152, no. 2, pp. 232–241, 2005.

[21] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.

[22] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the Society for Industrial & Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[23] M. Isard and A. Blake, "Condensationconditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[24] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 319–336, 2009.

[25] M. Teutsch, *Moving Object Detection and Segmentation for Remote Aerial Video Surveillance*. KIT Scientific Publishing, 2015, vol. 18.