Players Tracking in Football Game

André Ribeiro [1]

[1] IST, Technical University of Lisbon, Portugal
andrerr.ist@gmail.com

Abstract - *The analysis of human movement has been widely studied and has received special attention of researchers in the areas of image processing and vision. The fact that there is a wide spectrum of situations, such as tracking human figures in film sequences or determining the activity that is being held by the human (walk, run), makes the investment in research viable.*

In this thesis, we focus on the problem of the detection and the tracking of football players without considering the large variations in their silhouettes. It is quite a challenging task due to many difficulties such as player occlusion, similar player appearance, video blur and the emergence of evidences from external environment.

We propose a solution for player tracking which is based on a multiple object tracker, a particle filter named Condensation where each player is independently fitted to a model, and the sampling probability for the group of samples is calculated as a function of the fitness score of each player.

This solution is widely used in sports in which the relative position of the player within the team has to be determined as a time function. If this data is sufficiently accurate, a range of additional information can be determined, such as player's speed and distance accomplished.

Keywords - *Automatic Sample Detection, Particle Filter, Condensation, Multi-Part Model, Occlusion*

1. Introduction

Automated tracking of multiple objects is still an open problem in many settings including surveillance, sports and many others. A football game is a complicated activity, which involves a lot of interaction both between players, and the team as a whole. This interaction exhibits several challenging aspects such as significantly variation in players' shape, variable players' speed, sudden change in players' direction and players' occlusion.

This work aims to produce a tracker which is capable of, automatically, detect and track football players, identify their position as opposed to recognising whether they are walking, running, kicking the ball or involved in a set play.

The proposed solution for multiple players tracking is based on the Condensation/Particle Filter approach where each player is independently fitted to a model, and the sampling probability for the group of samples is calculated as a function of the fitness score of each player.

This paper is organized as follows: Section 2 describes background and foreground segmentation using colour spaces. Section 3 presents the particle filter based tracker and some improvements for the algorithm. Experimental Results are shown in section 4 and Discussion and future improvements are made in section 5.

2. Foreground and Background Segmentation

Before getting to the player tracking problem, we must face player detection. It is very unlikely to get good results in tracking if a good detection is not achieved. Moreover, if what we are trying to achieve is a completely automatic algorithm, it is important not to need human aid at the stage of player detection.

For that, and because football guidelines (produced by International Football Association Board) demand that the two teams use distinguish colours between them and also the referee and assistant referees, we will use colour spaces for foreground and background segmentation.

2.1. Choosing Colour Space

Colour spaces are mathematical models that describe how colours can be represented by numbers, usually with three or four variables. The more common spaces are RGB, HSV and YCBCR and we will choose among them the one that best fits our goals.

RGB is an additive colour model in which red, green, and blue light are added together in various ways to reproduce a broad array of colours. The name of the model comes from the initials of the three additive primary colours: red, green and blue.

HSV is not an additive colour model as RGB because all the information is encapsulated in three different variables. Hue is the wavelength within the visible-light spectrum at which the energy output from a source is the greatest, Saturation is an expression for the relative bandwidth of the visible output from a light source and Value is a relative expression of the intensity of the energy output of a visible light source (also known as Brightness).

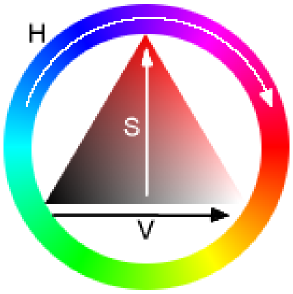


Fig. 1. HSV Colour space representation

YCBCR is not an absolute colour space, it is a way of encoding RGB information. Y is the luma component and can be stored with high resolution or transmitted at high bandwidth. Cb and Cr are the blue-difference and red-difference chroma components that can be bandwidth-reduced, sub sampled, compressed, or otherwise treated separately for improved system efficiency. Because of its efficiency as a representation of colour for storage and transmission, it is mainly used in digital photography.

Although the football videos used for tests were not sufficiently comprehensive, they allowed us to have an idea on the possible best colour space for later use.

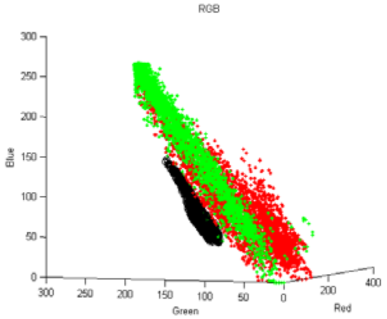


Fig. 2. Scatter plot of three dimensions of RGB

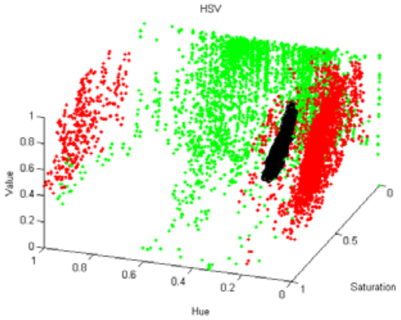


Fig. 3. Scatter plot of three dimensions of HSV

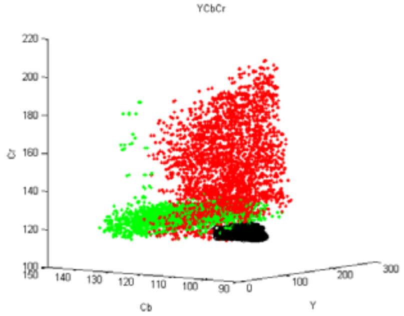


Fig. 4. Scatter plot of three dimensions of YCbCr

Various experiments were performed in a large number of frames. Our goal is to determine which colour space could better distinguish foreground from background pixels and the percentage of foreground pixels labelled as background. The performance of detection is shown in Table I.

TABLE I
PIXEL ERROR CLASSIFICATION (%)

	Video A	Video B	Video C	Mean
	Frame10	Frame10	Frame18	
		RGB		
Team1 Team2	3.52 0.89	17.67 12.99	14.96 5.01	9,17
		YCbCr		
Team1 Team2	1.23 0.08	0.35 0.37	4.21 0.09	1.06
		HSV		
Team1 Team2	0 0.02	0.26 0.29	4.21 0.12	0.82

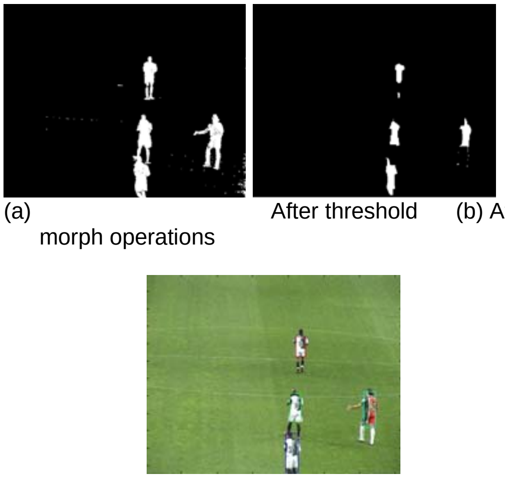
For example, in RGB colour space, the percentage of foreground pixels labelled as background in video B frame 10 for team 1 is 17.67%, while using HSV the percentage decreases to 0.26%.

The analysis of the results allows us to assume that HSV is the best model to use in this work.

2.2. Automatic Player Detection

For automatic player detection, an algorithm was developed which, using only a background sample chosen by the user, determines all the necessary data to tracking start.

To distinguish possible foreground from background pixels an initial threshold was made, resulting in fig.5 (a). This threshold results in 0 value for background pixels and value 1 to foreground pixels.


(a) After threshold (b) After morph operations

(c) Detected Players

Fig. 5. Steps for automatic player detection

After obtaining the threshold, a set of morphologic operations were performed to refine the detection result. These operations were made using the MatLab command "bwmorph" employing the options *clean*, *dilate* and *erode* sequentially. As it is possible to see in fig.5 (b) there are some bad detected foreground pixels, and therefore another improvement was applied. Thus, the median of height and width of the detected area was calculated, and only the areas within the size (incremented or decremented by a deviation) were taken into account.

3. TRACKING MULTIPLE PLAYERS

Most general tracking applications are suited for similar objects identification. This concept is not applicable when analysing the movement of football players because a single shape model looks unsuitable for modelling their silhouette. The approach taken in this work is just to fit a bounding box to each silhouette.

This bounding box is represented by four variables: the coordinates from the centre of the box **x** and **y**, the width **w** and the height **h**. Thus, a sample, representing an instance of a player, can be described as:

$$x = (x, y, w, h) \tag{1}$$

Being x_t^i an instance of a sample at time *t*, a particle can be formed as s_t^j along with the corresponding sampling probability π_t^j :

$$s_t^j = \left(x_t^1, x_t^2, \dots, x_t^{n_j}, \pi_t^j\right) \tag{2}$$

where n_j represents the number of players in the tracking process. A group of particles thus can be represented as:

$$S_t = \left(s_t^1, s_t^2, \dots, s_t^N\right) \tag{3}$$

where **N** is the user defined number of particles in the algorithm. An example is shown above:

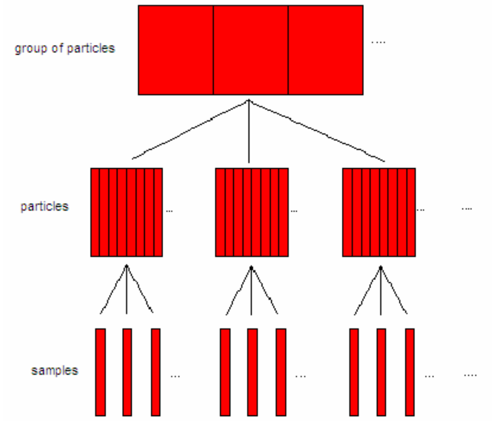


Fig. 6. Algorithm variable representation

3.1. Statistical Model

Following [1], for better representation of the sample probability we adopt a multi-colour observation model based on HSV colour histograms.

Although there are some disadvantages like ignoring form or texture information, colour histogram takes into account chromatic information, a feature strongly linked to a football game.

In order to decouple chromatic information, our colour model is obtained using histogramming techniques in HSV colour space. We start by populating the histogram with $N_h \times N_s$ bins, using $N_h = 10$ and $N_s = 10$. Thus, the first bin is filled with the pixels that belonging to the first Hue sampling interval, belong to the first Saturation sampling interval too.

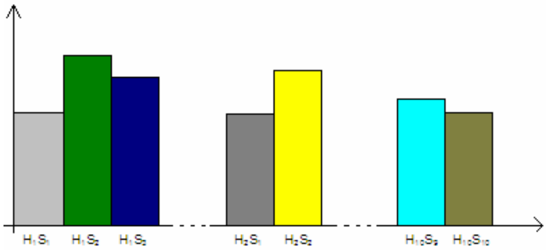


Fig. 7. HS Colour Histogram

At this point, only the Value component splitted into 10 sampling intervals $[\![(N]\!]_v = 10)$ remains for analysis. Here, no pixel combination is performed, counting only the number of pixels that fit in each Value sampling interval, adding it later to the upper histogram represented in Fig. 7. The resulting complete histogram is now composed of $N = N_h N_s + N_v$ bins:

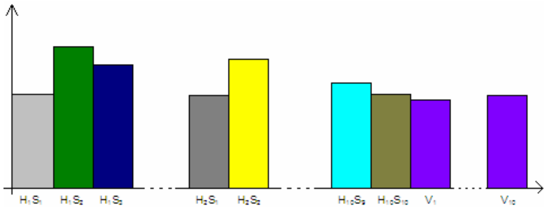


Fig. 8. HSV Colour Histogram

It is important to notice that, for later use, the histograms need to be normalized, imposing:

$$\sum_{n=1}^{N} q_t(n) = 1 \tag{4}$$

Each player sample is now represented as a histogram by binning all pixels. It lacks now to determine a function that compares the histograms. Among all the existent statistical distances, we choose to use the *Bhattacharyya distance*, defined by:

$$D[q_1, q_2] = \left[1 - \sum_{n=1}^{N} \sqrt{q_1(n) q_2(n)}^{\frac{1}{2}}\right] \tag{5}$$

where q_1 and q_2 are the histograms to compare and N the number of sampling intervals.

3.2. Particle Filter

Based on multiple object condensation tracker, our particle filter is divided in three distinct steps: Prediction, Filtering and Importance Sampling.

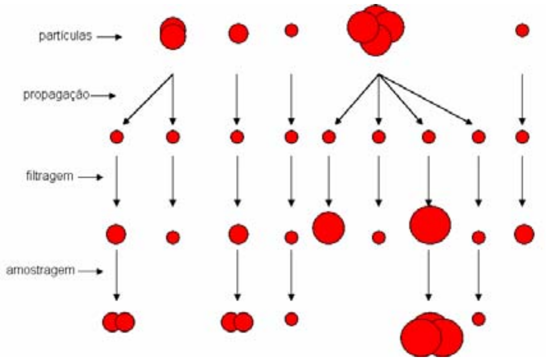


Fig. 9. Particle Filter representation

3.2.1. Prediction

As it is quite difficult to determine a model that represents the movement of players, our prediction model notices only the player current position $d = (x, y)$ and a completely random increment:

$$x_t = x_{t-1} + \varepsilon_x \tag{6}$$
$$y_t = y_{t-1} + \varepsilon_y \tag{7}$$

where $\varepsilon_x, \varepsilon_y \sim N(0, \sigma_1)$ and σ_1 assumed as 15pixels.

Beyond the estimation of player movement, it is also important to obtain a good approximation to the bounding box size because this value will have influence during the sample probability calculation. We have then:

$$w_t - f_w(y) \tag{8}$$
$$h_t = f_h(y) \tag{9}$$

with *w* as width, *h* as height and where $f_w(y)$ and $f_h(y)$ are initially calculated polynomials to linearize the bounding box size proportionally to the players *y* position.

3.2.2. Filtering

After predicting all the samples, our next goal is to calculate the probability of every sample and the particles populated by them.

This probability is calculated proportionally to the *Bhattacharyya distance* described in equation (5), resulting in:

$$p(x_t) = e^{-\lambda \times D[q^*, q(x_t)]^2} \tag{10}$$

Assuming that, as seen on [1], a good estimation for the value λ is 20, some tests were performed to ensure the veracity of this number. Varying the λ value from 0 to 50 it becomes obvious that the best results were obtained for values near 20.

After the calculation of the sample probability, it is now required the determination of the particle probability. Thus, for the particle probability we have:

$$\pi_t = \pi_{t-1} \times \prod_{k=1}^{K} p(x_t^k) \tag{11}$$

where *K* is the number of samples in each particle, in other words, the number of tracked players and *t* the analysed frame.

For later use in Importance Sampling, it is important to normalize all the particle probabilities:

$$\sum_{m=1}^{M} \pi_t^m = 1 \tag{12}$$

where *M* is the number of used particles in the algorithm.

We must realize that if the fitness score for each sample within the particle is similar then, the particle probability is increased but, on the other hand, if one or more samples have a poor fit then, the probability is reduced. This function also allows the reward of particles containing all reasonable samples rather than particles with quite dispar samples, some very good and some very bad.

3.2.3. Importance Sampling

Due to the normalization produced in (12) it is possible to define a range of values between 0 and 1 that allow us to do the importance sample step. Each range is built by:

$$x_i = \sum_{m=1}^{i} \pi_t^m \tag{13}$$

For better understanding this step, an example is shown in the following figure where we assume that $\pi^m \equiv \pi(m)$ is the probability of the particle m.

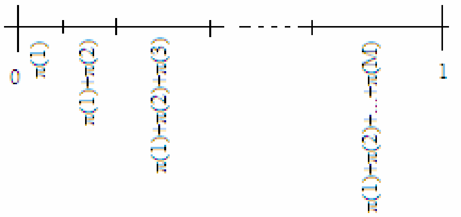
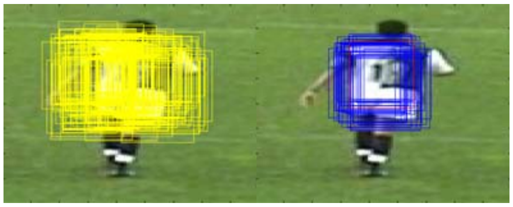


Fig. 10. Importance Sampling range intervals

To determine which particles are supposed to be propagated to the next frame, it is required to select M random numbers between 0 and 1. Based on these random values and the range intervals explained on Fig. 10 the propagated particles are selected. For instance, if the random number was in the second interval $[\pi(1)\ \pi(1) + \pi(2)]$, the selected particle for propagation would be the second.

By the example described above, it is easy to assume that the particles with the best overall score will have the biggest interval range and therefore, will be the most selected particle by the random numbers.

The particle with the highest probability is used as the best particle for the players' representation.



(a) Before Sampling (b) After Sampling

Fig. 11. Importance Sampling step – All particles in yellow; Selected particles in Importance Sampling in blue; Best particle for player representation in red

3.3. Multi-part Colour Model

If the tracked region contains patterns of different colours, e.g., the football shirts and shorts, the histogram model introduced before will capture them. However, all relative spatial arrangement of these patterns will be lost.

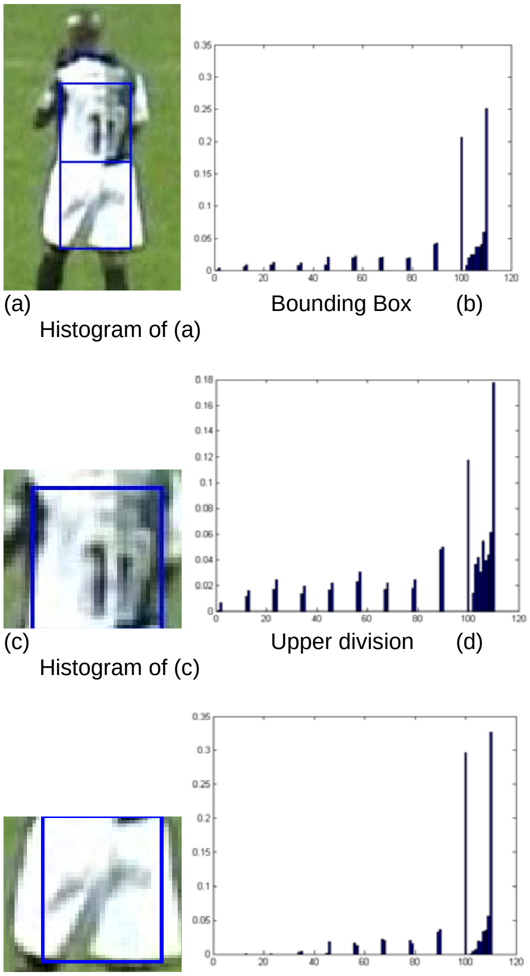
To improve the tracker performance, all these spatial arrangement should be kept. To achieve this, we propose the split of the tracked region into sub-regions with individual histograms models: $R(x) = \cup_{j=1}^{J} R_j(x)$.

We consider that these sub-regions are rigidly linked between them and so, the relative orientations are kept.

The new sample probability becomes as:

$$p(x_t) = e^{-\lambda \times \frac{1}{J}\sum_{j=1}^{J} D^2[q_j^*, q_j(x_t)]} \tag{14}$$

where $q_j(x_t)$ is the histogram of the sub-region $R_j(x_t)$ and *J* the number of sub-regions.



(a) Bounding Box (b) Histogram of (a)

(c) Upper division (d) Histogram of (c)

(e) Lower division (f) Histogram of (e)

Fig. 12. Multi Part Example – The splitting of the region of interest (a) into two distinct sub-regions (c) and (e)

3.4. Background Modelling

Until now, only foreground information is used in the algorithm and none of background pattern is taken into account.

As in our study the camera is fixed, the background views can be acquired offline so that the robustness of the tracker can be enhanced.

This important background information is thus added to equation (14) to achieve a new sample probability:

$$p(x_t) = e^{-\lambda x \frac{1}{J} \sum_{j=1}^{J} \left(D^2\left[q_j^*, q_j(x_t)\right] - D^2\left[q_j^-, q_j(x_t)\right] \right)} \quad (15)$$

where q_j^- refers to the reference background histogram.

Using extreme cases for explaining the applicability of equation (15) we will have two different situations.

Assuming that the tracked region is similar to the reference region, $D^2\left[q_j^*, q_j(x_t)\right] = 0$, and totally distinct from the background, $D^2\left[q_j^-, q_j(x_t)\right] = 1$, we have then $p(x_t) = e^{-\lambda x (0-1)} = e^{20} = 4{,}8517 \times 10^8$.

On the opposite situation, we will have a tracked region totally distinct from the reference, $D^2\left[q_j^*, q_j(x_t)\right] = 1$ and similar to the background, $D^2\left[q_j^-, q_j(x_t)\right] = 0$, resulting in $p(x_t) = e^{-\lambda x (1-0)} = e^{-20} = 2{,}0612 \times 10^{-9}$.

4. Experimental Results

This section shows detection and tracking results on football players in a video sequence. The experiments are performed using a frame rate of 25 frames per second, a window size of 720 horizontal pixels per 576 vertical pixels without zoom.

4.1. Players Detection

As told before, this is an important step for a good tracking process. For better analyses of this step, we started from a simple approach that consists on the detection of white points in a black background. Using MatLab, a 50x50 pixels window with white dots was created resulting in the detection above:

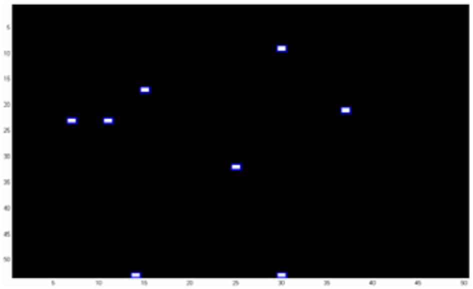


Fig. 13. Detection of white dots on a black background

After the success of this detection, the next step was based on a more complex environment. With the purpose to recreate the grass, the black background was replaced by a not homogeneous green background and, with the intention of recreates the two different teams, the white dots were replaced by coloured objects. This simulation had a window size of 640x480 pixels with objects sizing 60x20pixels, horizontal and vertical respectively. An example of this detection is shown in Fig.14.

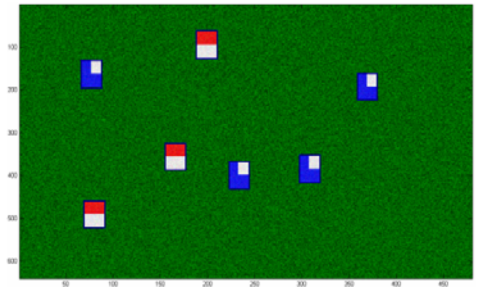


Fig. 14. Detection of coloured objects on a not homogenous green background

Finally, after running through all simulations environments, a test in a real environment was performed.

The good performance of this detection is illustrated in Fig. 15.

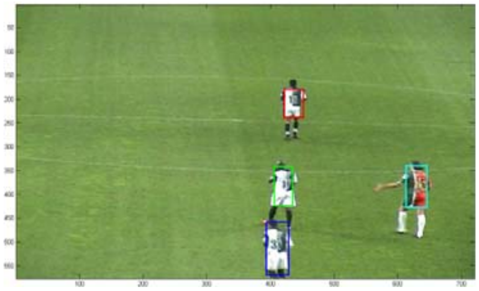


Fig. 15. Detection in real environment

During the tests some problems arose, which made less accurate the determining of the players position. Some of these problems include the emergence of evidences from external environments such as spectator's seats, players' bench or publicity boards around the field. Fig.16 shows an example of a not successful detection.

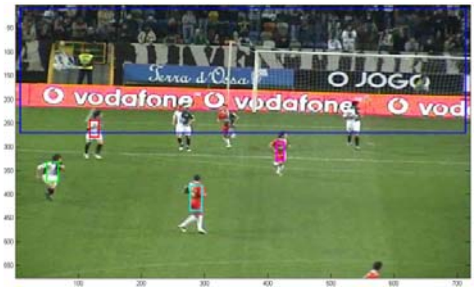


Fig. 16. Not successful detection

4.2. Players Tracking

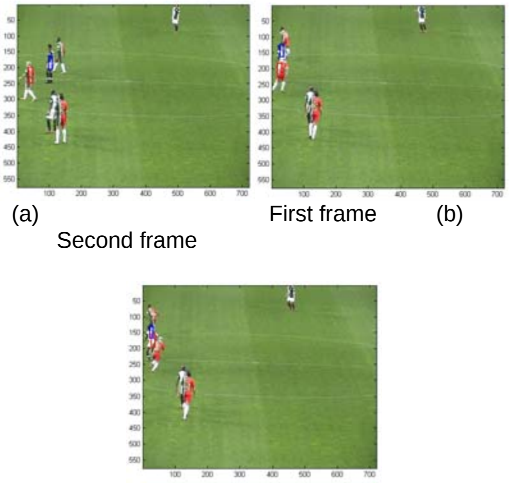
In an advanced phase of this work, where the goal was to test the behaviour of the tracker, the same three tests were performed but now using a random movement dynamic.

Having the same black background with white dots as described in the first example in section 4.1, we introduced a random movement factor of 2 pixels. Initially we tracked only one white dot, and after the successful of this procedure been confirmed, we've proceed the study by tracking 8 dots. Analysing the results from these tests, we found that the tracker has accurately determined the path of all 8 points under analysis.

As in section 4.1, we've continued the study with a more complex environment with a not homogeneous green background and two different types of coloured objects, adding a random movement factor of 5 pixels.

A first successful test was performed tracking only one object and, after that, we've proceeded tracking all the objects in the window and, once again, it was possible to determine the path made by each object.

When getting to real environment tracking, and acknowledging that no intrinsic information from the camera was available, neither the pitch size or the correspondence between the size and the video sequence, sometimes was difficult to recreate the players' path. Allied to these factors we sum the occlusion problem (presented in Fig. 17), similar player appearance, video blur and the emergence of evidences from external environments (shown in Fig. 18).

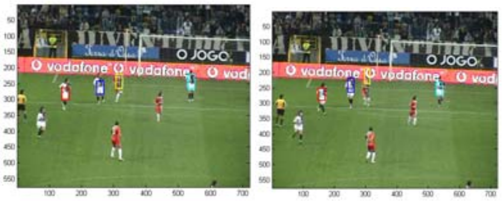


(a) First frame (b)

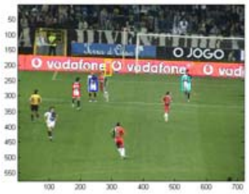
Second frame

(c) Third frame

Fig. 17. Occlusion error – Running well in the first frame, when facing an occlusion in frame 2 the tracker loses the players in frame 3.



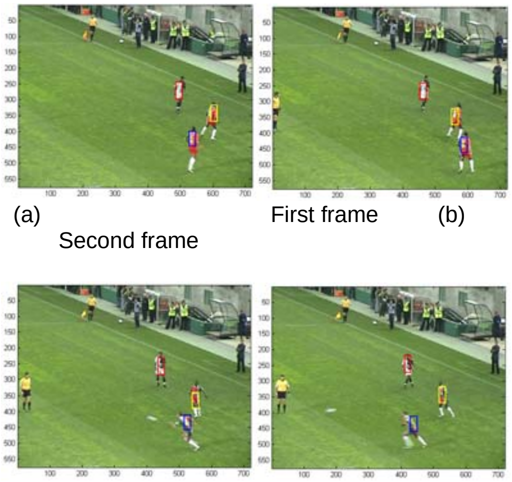
(a) First frame (b)
Second frame



(c) Third frame

Fig. 18. External elements – When the player passes through the publicity panels, the tracker gets confused.

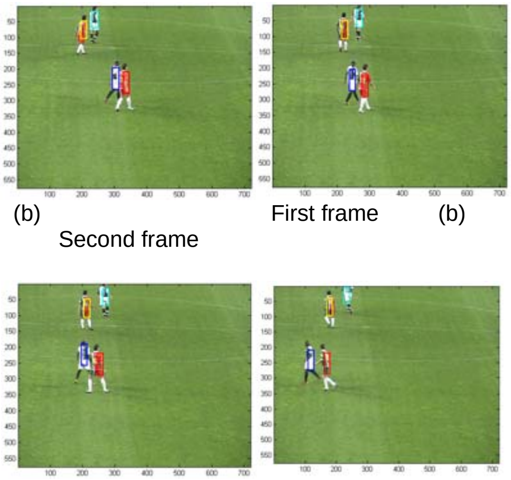
Despite of the above problems decrease the accuracy on defining the players path, it was most often possible to obtain good tracking results. As shown in Fig. 19 and Fig. 20, it was possible to track players in real environment.



(a) First frame (b)
Second frame

(c) Third frame (d)
Fourth frame

Fig. 19. Example of successful tracking



(b) First frame (b)
Second frame

(d) Third frame (d)
Fourth frame

Fig. 20. Example of successful tracking

A feature added to this work is the possibility to view the path performed by the players. As seen in Fig. 21, the path can become instable, mostly because the difference of 3 or 4 pixels doesn't influence the correct determination of player tracking.

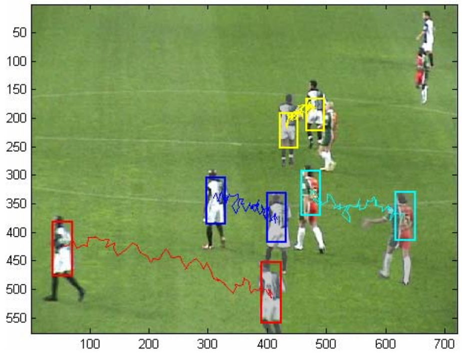


Fig. 21. Players' path representation

5. Discussion

We have described an approach to combine the object detection with the Condensation particle filter.

For both of the phases, three tests were performed to check the validation of the algorithm. The first two tests were based in simulation environments, where all factor are predictable. At the opposite side, the third and final test was made on real environment using video sequences where a lot of unpredictable situations were present.

In the first phase, where our goal was achieved a good players' detection in the initial frame selecting only a background sample, the first test allowed us to obtain the correct position of the white dots on a black background (Fig. 13). The second test shows us that it is possible to detect coloured objects in a not homogeneous background (Fig. 14), capturing each coloured object individually. Running the final test, where no priori information was available, we concluded that the detection is somehow susceptible to errors if the presence of external objects is noticed. Despite of these errors, if the background is reasonable homogeneous the obtained results can be quite satisfactory (Fig. 15).

Further improvements such as the correspondence between the detected samples and each team in the field, can be obtained by including à priori information about the equipment colour in all perspectives (frontal, back and side view).

In the second phase, consisting on tracking players, the main difficulties were present in occlusion scenes and sudden changes of players' direction. As the second factor doesn't happen often, we focused our analysis mainly on the video sequences with occlusion. Within the occlusions we could differentiate two kinds: occlusions between players from the same team and occlusions between players from different teams (Fig. 17). Referring to the first kind of occlusions, our algorithm is robust and capable of solving it. It is on the second kind that lays some of our problems because the players have the same chromatic information which makes the disambiguation more difficult. Having access to different perspective videos from the same sequence, could be an improvement because it would be unlikely not to detect a player not visible in the first video capture.

Another future research perspective that would allow more robustness in the algorithm will pass through the improvement of the Multi Part Colour Model (Section 3.3), because if a player silhouette varies a lot during a football game, it is reasonable that the algorithm covers this situation.

The results show that most players are successfully detected and tracked even if sometimes an occlusion is present (Fig. 20).

We must realize that if we fail in the detection process it is unlikely to have a good tracking result. In practice the results obtained from the detection phase, are used in the players' tracking algorithm.

During the tests, it was not possible to calculate a hit percentage or scale results because, being the player a not homogenous mass, it is impossible to determine a univocal result for his position.

Referring to players' real position on the pitch, as we were not provided the intrinsic parameters of the cameras, it became unviable to calculate it. This feature was not a study goal because it doesn't initially fit our objectives.

With this work, our research perspectives concerning to the detection and tracking of football players were accomplished.

6. References

[1] Pérez, P., Hue, C., Vermaak, J., Gangnet, M., "Color-Based Probabilistic Tracking", European Conference on Computer Vision, 2002.

[2] Okuma, K., Taleghani, A., de Freitas, N., Little, J. J., Lowe, D. G., "A Boosted Particle Filter: Multitarget Detection and Tracking", European Conference on Computer Vision, 2004.

[3] Needham, C. J., Boyle, R. D., "Tracking Multiple Sports Players Through Occlusion, Congestion and Scale", British Machine Vision Conference, Report 2001.07.

[4] Xu, M., Lowey, L., Orwell, J., "Architecture and Algorithms For Tracking Football Players With Multiple Camaras", IEE Proceedings Vision, Image and Signal Processing, 2005.

[5] Ren, J., Orwell, J., Jones, G. A., "Estimating the Position of a Football from Multiple Image Sequences", Digital Imaging Research Center, Kingston University, CiteSeer.

[6] Bebie, T., Hanspeter, B., "SoccerMan - Reconstructing Soccer Games from Video Sequences", Conference on Image Processing, 1998.

[7] 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, 2002.

[8] Foley, J., Van Dam, A., Feiner, S., Hughes, J., "Computer Graphics: Principles and Pratice", Addison-Wesley, 1990.

[9] Gonzalez, R. C., Woods, R. E., Eddins, S. L., "Digital Image Processing using Matlab", Prentice Hall, 2004.

[10] D. Comaniciu, V. Ramesh, P. Meer, "Real-Time Tracking of Non-Rigid Objects using Mean Shift, To appear, IEEE Conf. on Comp. Vis. and Pat. Rec., Hilton Head Island, South Carolina, 2000.

[11] J. Liu, X. Tong, W. Li, T. Wang, and Y. Zhang, "Automatic player detection, labeling and tracking in broadcast soccer video," accepted by BMVC 2007.

[12] Borg, J., Master Thesis in Computer Vision "Detecting and Tracking Players in Football Using Stereo Vision", Linköping University, Sweden, 2007.

[13] Toyama, K., Hager, G. D., "Tracker Fusion for Robustness in Visual Feature Tracking", Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference, 1995.