archives-ouvertes.fr

Adaptative Tracking of Non Rigid Objects Based on Color Histograms and Automatic Parameter Selection

Aude Jacquot, Peter Sturm, Olivier Ruch

To cite this version:

Aude Jacquot, Peter Sturm, Olivier Ruch. Adaptative Tracking of Non Rigid Objects Based on Color Histograms and Automatic Parameter Selection. IEEE Workshop on Motion and Video Computing, Jan 2005, Breckenridge, United States. IEEE Computer Society, pp.103–109, 2005, <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4129592>. <10.1109/ACVMOT.2005.19>. <inria-00524393>

HAL Id: inria-00524393
https://hal.inria.fr/inria-00524393

Submitted on 25 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adaptive Tracking of Non-Rigid Objects Based on Color Histograms and Automatic Parameter Selection

A. Jacquot, P. Sturm
INRIA Rhône Alpes
Grenoble, FRANCE

O. Ruch
Thales Optronique
Guyancourt, FRANCE

Abstract

One of the main difficulties in visual tracking is to take into account appearance changes (not only of the target but also of or due to the scene, illumination for example). The use of a Bayesian framework is very flexible and has proven to be very efficient in visual tracking. Moreover, color or greylevel histograms allow to track an objet with a low computational cost. The recently proposed color-based trackers integrated in a probabilistic framework [1, 3] are efficient for a given application (face tracking for example) but can not be generalized easily, due to the initialization and the adjustment of the different tracker parameters that are dependent on the input sequence. This paper presents a method based on color integrated in a particle filter that allows to cope with some of the usual problems of visual tracking (occlusions, target appearance changes, changes in resolution or in illumination) and to adapt easily to different applications (tracking of structures in aerial imagery as well as football players). The novelty of the tracker is its ability to automatically regulate all the parameters needed for tracking, which makes it flexible and easily usable for different applications.

1. Introduction

Whatever the object we want to track, tracking is based on some model describing its appearance: this model can include prior information on the target as well as information extracted from the previous frames in the sequence. The model can contain geometric contours, image patches, global descriptors or other features. One of the main factors that limits the performance of visual tracking algorithms is the lack of a suitable appearance model for the target. Template matching methods can not directly cope with appearance changes and motion estimation based methods allow the appearence model to change rapidly but tend to drift away from targets.

This paper proposes a robust appearance model for tracking using color distributions. Histograms are robust to partial occlusion, rotation and have a low computational cost. Furthermore, particle filters have proven to be efficient and reliable in cases of clutter and occlusions. Several trackers based on color histograms integrated in probabilistic frameworks have been proposed recently [1, 2, 3]; to the best of our knowledge these algorithms are efficient for a given application but can not be adapted easily to another target. Other techniques proposed by Bradski (the "Camshift" [4]), Comaniciu (the "Mean Shift" [5]) or more recently Zivkovic (who proposes an extension of the mean shift algorithm in [7]) do not use probabilities but make a deterministic search of the region whose color content best matches the reference model. These methods have the same limitations as the previous ones. The novelty of the proposed tracker lies in the integration of some criteria which allow to automatically select the number of bins of the histogram needed for the tracking and in a new way to update the model. Our tracker is robust to occlusions, changes in illumination as well as changes of appearance of the target, and has shown to be efficient in tracking objects with a hand-held camera as well as tracking buildings or static structures in aerial imagery.

The outline of this paper is as follows: in Section 2 we briefly describe the basic method: particle filtering, the way to use color for tracking and the way to integrate both. Section 3 describes the improvements we have made: the gain of spatial information obtained by dividing the patch of interest, the model update and the automatic selection of the number of bins of the model's color histogram and parameters needed for the tracking. Finally some results are presented in Section 4.

2. Color-based probabilistic tracking

The aim of this section is to present the basis of the tracking of non rigid objects using color histograms in conjunction with a probabilistic framework [1, 3].

2.1. Recalls on particle filtering

We use the Bayesian framework to track objects in the case where the posterior density $P(X_t \mid Z_t)$ and the observation model $P(Z_t \mid X_t)$ are not necessarily Gaussian. The object tracked is characterized by its state vector X_t, and the observations up to time t are defined by the vector Z_t.

The idea of particle filtering is to approximate the probability distribution of the object state by a weighted sample set. Each sample is an element which represents the hypothetical state s of the object, associated with a weight π. The sample set can be written as: $S = \{(s^{(i)}, \pi^{(i)}), i = 1, ..., n\}$ where $\sum_{i=1}^{n} \pi^{(i)} = 1$

The evolution of a sample set is given by propagating each sample according to a system model (here a motion model, see below). Each element of the set is then weighted in terms of the observations, and the mean state of the object is estimated at each step as:

$$E[S] = \sum_{i=1}^{n} \pi^{(i)} s^{(i)} \quad (1)$$

One of the advantages of particle filtering is that it models uncertainty, thus making it more robust in case of occlusion or clutter.

2.2. Color histogram as a model

As said previously, we use color distributions to model our target because of their robustness to rotation, partial occlusion and non rigidity of the target. Suppose the distributions are discretized into K bins (see Section 3.1 for the automatic selection of K). In our approach, we model the target by an ellipsis. The histograms can be calculated in RGB or any other color space, or simply in grey level space, depending on the input sequence. To partially cope with the loss of spatial information when using histograms, Nummiaro [1] and Pérez [2] assigned different weights to the pixels of the ellipsis to increase the reliability of the color distributions; smaller weigths are given to the pixels far away from the ellipsis center, using the following weighting function:

$$k(x) = \begin{cases} 1 - x^2 & \text{if } x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

where x is the distance from the ellipsis center. Note that other weighting functions can be used: Comaniciu, for example in [5], uses the Epanechnikov kernel.

A color or greylevel distribution $p_x = \{p_x^{(j)}\}_{j=1,\ldots,K}$ at a location x is calculated as

$$p_x^{(j)} = C \sum_{x_i \in E} k\left(\frac{\|x - x_i\|}{\sqrt{(l_x^2 + l_y^2)}}\right) \delta\left(h(x_i) - j\right) \quad (2)$$

where δ is the Kronecker delta function, E is the set of pixels in the ellipsis, l_x and l_y are the ellipsis half lengths, $h(x_i)$ assigns one of the K bins of the histogram to a given color at location x_i and C is the normalization factor, which ensures that $\sum_{j=1}^{K} p_x^{(j)} = 1$. The expression of C is given by

$$C = \left[\sum_{x_i \in E} k\left(\frac{\|x - x_i\|}{\sqrt{(l_x^2 + l_y^2)}}\right)\right]^{-1}$$

The similarity of two distributions p and q is measured by the Bhattacharyya coefficient

$$\rho[p, q] = \sum_{j=1}^{K} \sqrt{p^{(j)} q^{(j)}} \quad (3)$$

For two identical distributions, we have $\rho = 1$, which corresponds to a perfect match. We use the Bhattacharyya distance $d = \sqrt{1 - \rho[p, q]}$ in our algorithm.

2.3. How to combine color histograms and particle filtering?

We want to track a patch of interest in the image plane. We choose to parameterize this patch by an ellipsis

$$s = \left\{x, y, \dot{x}, \dot{y}, \theta, l_x, l_y, \dot{l}_x, \dot{l}_y\right\}$$

where x and y represent the ellipsis center, $\dot{x}$ and $\dot{y}$ the velocities of the center, θ the ellipsis orientation, l_x and l_y the lengths of the ellipsis half axes, and $\dot{l}_x$ and $\dot{l}_y$ the velocities of l_x and l_y. This model is flexible in that the ellipsis parameters can vary independently.

To propagate the sample set, we use a first order model given by

$$s_t = A s_{t-1} + b_{t-1} \quad (4)$$

where b_t is a multivariate Gaussian random variable and A is a matrix designed in order to describe an object moving with constant velocity for x, y, l_x and l_x.

The tracker works as follows: in the first image the model distribution is calculated, and the set of particles is initialized. Then, for each image of the input sequence, we propagate the set of particles using the dynamic model previously defined. For each sample of the set, the Bhattacharyya distance between the model distribution and the sample distribution is calculated and used to compute the weight π of the sample. The weight associated to each particle of the set favors samples whose color distributions are similar to the target model. The weights are calculated using

$$\pi^{(i)} = \gamma \exp(-\beta d^{(i)}) \quad (5)$$

for each particle i of the set, where γ and β are some fixed constants and $d^{(i)}$ represents the Bhattacharyya distance between the i^{th} particle and the target model.

The last step is to re-sample the particles to ensure the efficiency of the evolution, and to determine the mean state of the object. During the re-sampling step, the particles are eliminated or duplicated according to their weight: the higher the weight of a particle, the more likely it is to be duplicated. Different methods exist, we chose to use a systematic re-sampling [9]. It consists in dividing the interval $[0, 1]$ into n segments. Then a uniform random variable U is generated on $\left[0, \frac{1}{n}\right]$; we define $U_1 = U$ and $U_i = U_{i-1} + \frac{1}{n}$ for $i = 2, .., n$. If U_i belongs to the j^{th} segment, then we pose $\Xi^i = X^j$, where Ξ^i is an element of the new sample set. The advantage of this method is that only $O(n)$ comparison tests are needed to produce the new sample set.

3. Our contributions

Many approaches have been proposed for tracking objects (with or without shape deformations) based on color histograms integrated in a probabilistic framework as described in the last section. But none of them are flexible enough to automatically regulate all the parameters in order to make them easily usable for different applications. We propose in this section some criteria that allow to automatically determine the tracking parameters.

3.1. The appropriate number of bins

The number of bins in our histograms is a crucial parameter and should be determined automatically. Too many bins in a histogram do not cope with changes in illumination or in the model appearence and most of the time the algorithm drifts away from the target. On the opposite, too few bins do not allow a good discrimination of the target, and the tracking fails. This evidence has been confirmed by our experiments, as shown in Section 4.

In most of the existing approaches, the number of bins seems to be chosen arbitrarily and kept fixed during the tracking. Nothing indicates that such a partition is optimal given the n-sample density we want to estimate. If we could find the optimal partition, the tracker should be more robust.

There have been many attempts in the past to solve the problem of determining the optimal number of bins from the data. Generally these methods are based on some asymptotic considerations. The problem with these approaches is that they do not perform very well in the case of small sample sizes due to their asymptotic nature. Moreover, many of them assume some prior information about the density. Recently, Birgé and Rozenholc [10] have generalized Akaike's estimator. Akaike's theorem is a statistical measure for model selection which states that if two models fit the data equally well, the simpler model will usually predict better. In the following, we briefly summarize their method of determining the optimal number of bins of our histograms. For the theoretical arguments underlying it, refer to [10].

The purpose is to find a histogram estimator $\hat{f}$ based on some partition $\{I_1, ..., I_K\}$ of $[0, 1]$ into K intervals of equal length. $X_1, X_2, ..., X_n$ are n samples from the unknown density f we want to estimate. K is given by

$$K = \arg\max_K (L_n(K) - penalty(K)) \qquad (6)$$

where $L_n(K)$ is the log-likelihood of the histogram with K bins, given by

$$L_n(K) = \sum_{j=1}^{K} M_j \log(\frac{KM_j}{n}) \quad \text{with} \quad M_j = \sum_{i=1}^{n} \mathbf{1}_{I_j}(X_i)$$

where $\mathbf{1}_{I_j}$ is the indicator function defined by

$$\mathbf{1}_{I_j}(x) = \begin{cases} 1 & \text{if } x \in I_j \\ 0 & \text{otherwise} \end{cases}$$

The penalty function is given by

$$penalty(K) = K - 1 + (log(K))^{2.5} \quad \text{for } K \geq 1$$

This approach is thus a typical example of model selection methods, making a compromise between the complexity of the model and its fidelity to the data.

3.2. Incorporating spatial information

The problem with the use of histograms is that all spatial information is lost, as opposed to templates, which use the whole spatial information. As said previously, assigning different weights to the pixels of the ellipsis according to their distance to the center allows to integrate some spatial information. We wanted our tracker to work with more spatial information. We decided to divide our ellipsis in four quarters, and proceed as previously explained for each one of the ellipsis quarters. This division of the ellipsis increases the robustness of the tracker since we have four measures of similarity between a hypothesis and the model that can be combined easily and allows a better discrimination between the object and the rest of the scene.

Another advantage of dividing our ellipsis is to use the criterion for the automatic selection of the number of bins for each one of the quarters. The number of bins can be different in each quarter of the ellipsis according to the amount of data available (an ellipsis quadrant containing an homogenous region does not need as many bins as a highly textured region).

Finally, the division of the ellipsis into quarters makes it easier to handle or detect partial occlusions: if the Bhattacharyya coefficient is bad for one of the quarters but very good for the others then a partial occlusion is detected. To determine the mean state of the object, we combine the four measures by calculating their median value.

3.3. The model update

The apparent color of an object can vary over time due to changes in illumination, in camera parameters or in object motion. To deal with these appearance changes, the model has to be updated. Particle filtering has already been used with static [8] or adaptive [6] models. Most of the time, the model is updated for each frame where the probability of the mean state is above a threshold fixed arbitrarily at the initialization. The risk with this method is to gradually drift from the target.

The idea we propose is the following: why should we update the model when it is still good? The model should only be updated when its appearance changes too much. So we use the following criterion: if the mean state is under a threshold π_T (see next paragraph for its setup), then we update the model using the following equation:

$$p_{E[S_t]} \leq \pi_T \Rightarrow q_t^j = (1-\alpha)q_{t-1}^j + \alpha p_{E[S_t]}^j \qquad (7)$$

Setting up of the threshold π_T: This threshold depends on different parameters more or less connected: the ellipsis size and the number of bins of the histograms. In fact, the larger the ellipsis is, the larger in general the number of bins will be. This results in a lower Bhattacharyya coefficient and the threshold for the model update should be lower too. To set it up automatically, we make the hypothesis that in the beginning of the sequence (the first images) the mean state of the object is well estimated and we set the threshold empirically: $\pi_T = \rho - c$ where c is a fixed constant.

The global scheme of the algorithm is given on Figure 1.

1. Initialization:
   - Selection of the ellipsis in the first image
   - Automatic determination of the number of bins of each histogram according to equation (6)
   - Computation of the model distribution in each ellipsis quarter with equation (2)
   - Initialization of the sample set
2. For each new image:
   - Propagate the sample set using the dynamic model according to equation (4)
   - For each sample of the set S_t, compute
     - The color distribution with equation (2)
     - The Bhattacharyya coefficients with (3)
     - The weights with equation (5)
   - Estimate the mean state according to equation (1)
   - Update of the target model if necessary with (7)
   - Re-sampling step (see Section 2.3)

Figure 1: The algorithm.

4. Experiments

This section evaluates the performances of our tracker. We first show how performant the automatic selection of the number of bins is, then we present the experimental results of our tracker on various sequences. The description of the sequences is made in Section 4.2.

4.1. Evalutation of our contributions

To give experimental evidence that our contributions lead to better tracking, we ran our algorithm several times (15) with the automatic selection of the numbers of bins and with fixed number of bins in the case of the tracking on the entire ellipsis. We use the mean value for our results, presented in Table 1; the numbers indicate the image number from which the tracking fails and bold numbers correspond to the results for the automatically chosen number of bins. To measure that the tracking fails, we established a "ground truth" of each one of the sequences by keeping in a file the coordinates of a good tracking. Tracking was then declared as failed if for one image, the size of the ellipse is too large or if the ellipsis center is too far away from the ground truth. The results show that fixing arbitrarily a number of bin can't be a good solution for tracking various objects. But the automatic selection leads most of the time to the optimal tracking.

# of bins	Football	Indoor	License plate
2	16	19	31
4	87	23	46
6	107	111	55
8	109	125	41
10	95	172	60
12	118	169	94
14	90	**168**	92
16	121	164	84
18	**117**	132	**93**
20	111	217	61
22	75	108	78
24	88	135	62
26	72	90	63
28	86	107	73
30	76	78	74
32	69	85	82
34	102	70	61
36	92	92	86
38	102	76	69
40	87	59	91

Table 1: Performances of the tracker using the entire ellipsis with various numbers of bins.

Figure 2 presents the results obtained for each sequence with the automatic selection of the number of bins, in the case of a tracking with the entire ellipsis or the 4 quadrants. The graphs shows that using 4 quadrants improve largely the performances of our tracking.

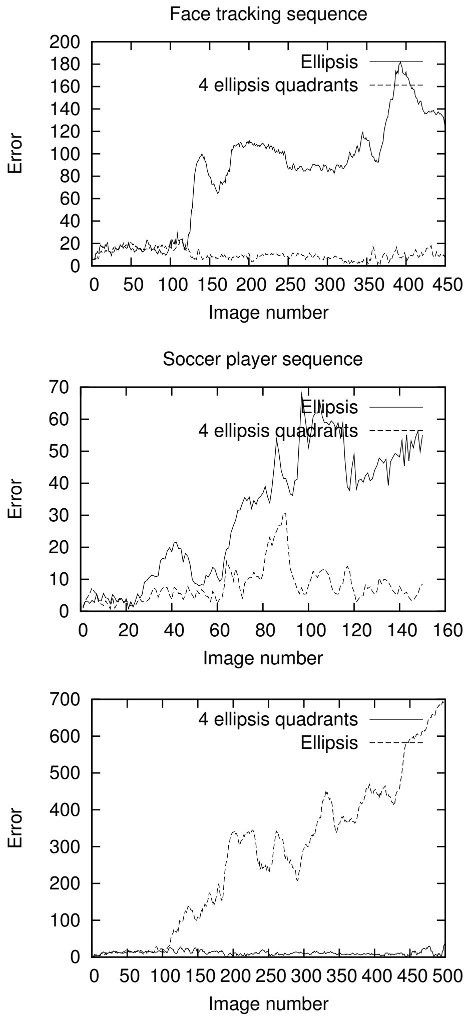

Figure 2: Influence of the ellipsis division into quadrants for the 3 sequences.

4.2. Tracking results for different sequences

We tested our tracker on different types of input: the results show that our approach allows us to use it efficiently for various applications independently of parameter initialization. Three sets of results are provided; in each experiment grey level sequences are used.

1. *Face tracking*: The sequence presents a person entering and moving around a room

Figure 3: Face tracking results: images 1, 140, 395, 412 and 449 of the sequence.

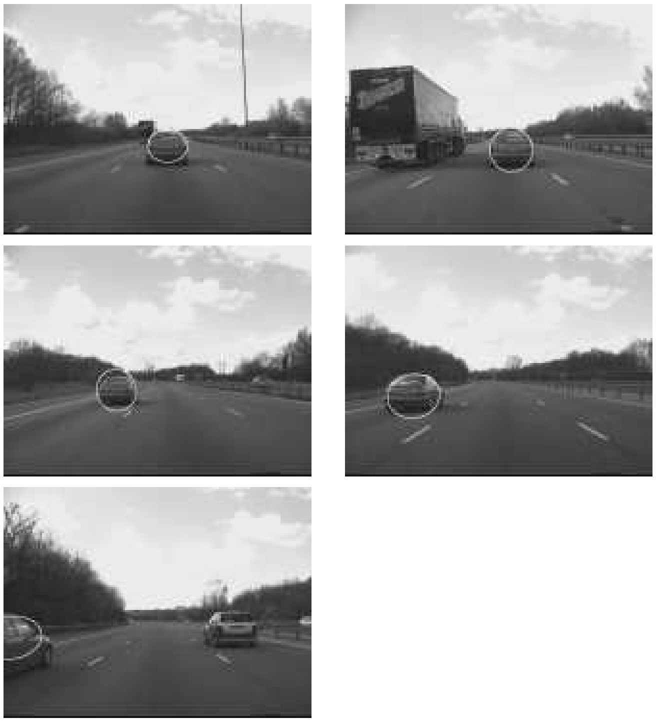

Figure 4: Car tracking results: images 1, 187, 347, 462 and 563 of the sequence.

Figure 5: License plate tracking results: images 1, 187, 347, 417, 520 of the sequence, and a zoom made on image 417.

(http://www.ee.oulu.fi/~mikak/tracking/FaceColor.html). There are some important appearance changes since the person turns around 360° during the sequence. The results show that the way to update the model in order to cope with appearance changes of the target is efficient. Furthermore, the algorithm is able to track even in the cases where the person moves with changes in speed.

2. *Car tracking*: The sequence is one of the PETS 2001 database (ftp://pets2001.cs.rdg.ac.uk/) in the context of a driver assistance application. We made two experiments, the results are presented in Figures 4 and 5: the particle filter has to deal with rapid movements of the target and the camera. The difficulties of this sequence are the different viewing angles of the tracked car, the changes in scale and the out of plane rotations; but we can see the good performance of our algorithm during the whole sequence. For the first experiment we tracked the entire car, achieving equivalent results to [1]. For the second one, we only tracked the rear license plate of the car. The additional difficulty of this experiment is the small size of the object to be tracked. The results presented in Figure 5 show that our tracker is able to track efficiently even small objects.

3. *Football player tracking*: The sequence is taken from a football match; the difficulties of this sequence are the fast motions of the players and the occasional occlusion of some players by others players on the ground. In frame 78, a player is falling down and another player attempts to catch the ball so a player is entirely occluded. Our tracker remains efficient even in this case.

5. Conclusions and discussion

These results suggest that our system is able to track:

- an object with large appearance changes such as shape and/or orientation changes.
- an object in a scene with scale changes.
- an object that moves with varying velocity as illustrated by the football player sequence.
- a deformable object (the football player for instance).

We also tested our algorithm on aerial sequences; the results show the robustness of our tracker for various applications.

The proposed tracker adds a criterion which allows to detect the optimal number of bins needed for the histograms in order to achieve robust tracking of various objects. Moreover we set up rules allowing the algorithm to work automatically for various applications. Our approach is a step towards a fully automatic and adaptative tracking. The tracking algorithm is based on color distributions integrated in a probabilistic framework. The experiments show that the tracker is robust to partial and complete occlusions, to appearance changes of the target as well as changes in illumination of the scene. Furthermore, the division of the ellipsis into smaller regions increases the robustness of the tracker.

Notice that we divided the ellipsis into quarters; it would be interesting to set up a criterion similar to the one used to select the number of bins in order to find the best compromise between the amount of spatial information in the model and the flexibility of histograms. Furthermore, we think about updating the number of bins during the sequence if appropriate (for example if the target grows in the image, updating the model with more bins in order to profit from the increasing of information). Also, the model update using the color distributions at the current mean state is expected to run into problems if a similar looking object is nearby; we work at solutions for this problem.

References

[1] K. Nummiaro, E. Koller-Meier, L. Van Gool, "Color Features for Tracking Non-Rigid Objects", *ACTA Automatica Sinica*, 2003.

[2] P. Pérez, C. Hue, J. Vermaak, M. Gangnet, "Color-based probabilistic tracking", *In European Conference on Computer Vision*, LNCS 2350, pp 661-675, Copenhagen, Denmark,2002.

[3] A.D. Jepson, D.J. Fleet, T.F. El-Maraghi, "Robust Online Appearance Models for Visual Tracking", *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Vol I, pp 415-422, 2001.

[4] G.R. Bradski, "Computer vision face tracking as a component of a perceptual user interface", *In Workshop on Applications of Computer Vision*, pp 214-219, Princeton, NJ, 1998.

[5] D. Comaniciu, V. Ramesh, P. Meer, "Kernel-Based Object Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 25, no 5, 2003.

[6] K. Nummiaro, E. Koller-Meier, L. Van Gool, "Object Tracking with an Adaptative Color-Based Particle Filter", *Image and Vision Computing*, 2002.

[7] Z. Zivkovic, B. Krose, "An EM-like algorithm for color-histogram-based object tracking", *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2004.

[8] T. Heap, D. Hogg, "Wormholes in Shape Space: Tracking through Discontinuous Changes in Shape", *Sixth International Conference on Computer Vision*, Bombay, India, 1998.

[9] F. Legland, "Filtrage particulaire", *Proceedings 19ème Colloque GRETSI sur le Traitement du Signal et des Images*, VolI, pp 1-8, Paris, 2003.

[10] L. Birgé, Y. Rozenholc, "How many bins should be put in a regular histogram", *Technological Report, Laboratoire Probabilités et Modèles Aléatoires, Université Pierre et Marie Curie*, Paris, France, PMA-721, 2002.

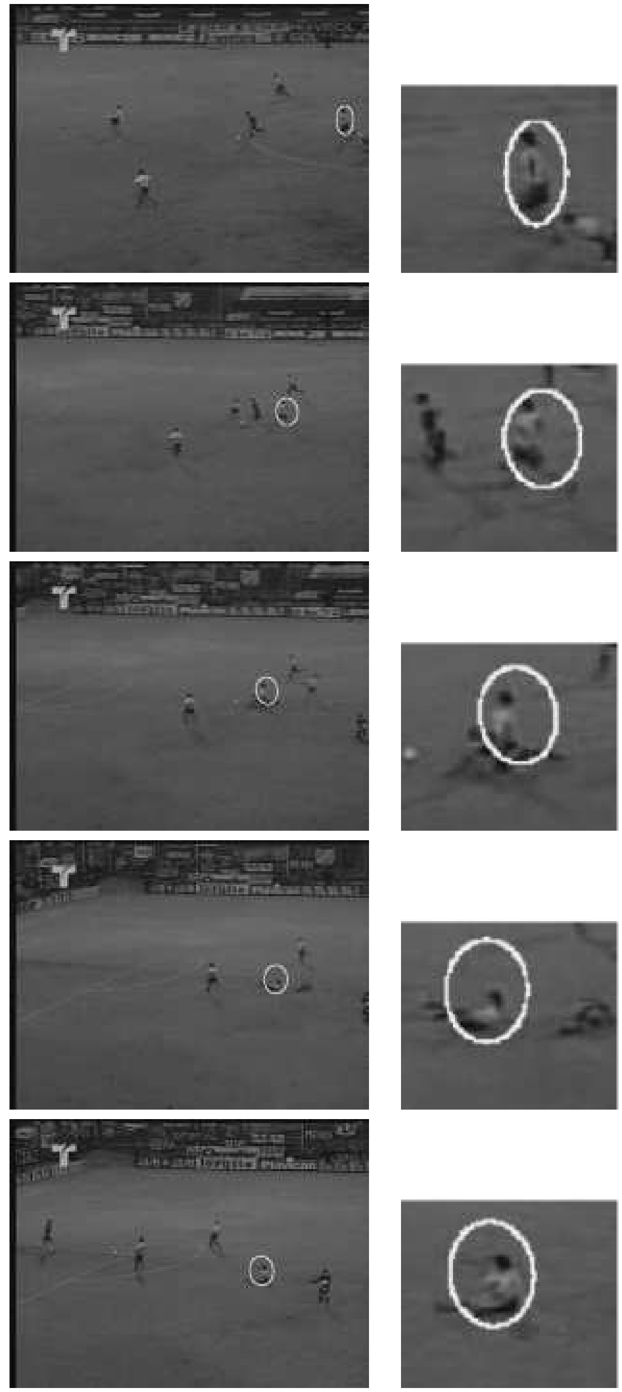

Figure 6: Football player tracking results: images 1, 55, 76, 97 and 141 of the sequence. In frame 76, the white player occludes the black one.