Probabilistic Color-Based Multi-Object Tracking with Application to Team Sports

Nicolas Gengembre, Patrick Pérez

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Probabilistic Color-Based Multi-Object Tracking with Application to Team Sports

Nicolas Gengembre — Patrick Pérez

N° 6555

Mai 2008

Thème COG

ISSN 0249-6399 ISRN INRIA/RR--6555--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
RENNES - BRETAGNE ATLANTIQUE

Probabilistic Color-Based Multi-Object Tracking with Application to Team Sports

Nicolas Gengembre , Patrick Pérez

Thème COG — Systèmes cognitifs
Équipe-Projet Vista

Rapport de recherche n° 6555 — Mai 2008 — 26 pages

Abstract: This paper addresses the problem of tracking multiple non rigid objects — such as humans — in videos. Firstly, it aims at showing how tracking can be improved with the help of background analysis. Background color modeling is used to optimise the target discrimination, to detect specific situations such as occlusions or clutter, and to perform selective adaptation of the target model. Background motion due to camera pan, tilt and zoom is estimated and compensated in the tracking procedure. This tracking procedure is based on particle filtering, so that occlusions are also dealt with. Another key feature of the proposed tracking algorithm lies in its extension to multiple objects, possibly with similar appearances. The capacities of these new developments are demonstrated on sequences with both zooms and occlusions, including occlusions between similar objects. Applications to team sports are especially demonstrated.

Key-words: Tracking, Multi-object tracking, Background analysis, Bayes classification, Color histogram, Camera motion, Interest points, Particle filter

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex
Téléphone : +33 2 99 84 71 00 — Télécopie : +33 2 99 84 71 71

Suivi probabiliste multi-objets basé couleur - Application aux sports d'équipes

Résumé : Cet article aborde le problème du suivi d'objets multiples non rigides — tels que des êtres humains — dans des vidéos. Il vise tout d'abord démontrer comment le suivi peut être amélioré grâce une analyse de l'environnement des objets (le fond). Une modélisation des couleurs du fond est utilisée pour l'optimisation du pouvoir discriminant du modèle de la cible suivie, pour détecter des situations spécifiques telles que la présence d'occultations ou de fouillis, ainsi que pour effectuer une adaptation selective du modèle de l'entité suivie. Le mouvement de fond, du aux mouvements de la caméra, est estimé et compensé dans la procédure de suivi. Celle-ci repose sur une approche probabiliste qui permet en outre de prendre en compte les occultations. Un autre point clé de l'algorithme proposé est son extension au problème du suivi d'objets multiples, éventuellement d'apparences similaires. Le potentiel de ces nouveaux développements est démontré sur différentes séquences, incluant la fois des problèmes de zoom et d'occultations, y compris d'occultations entre objets similaires. Des applications relatives aux sports d'équipes sont particulièrement étudiées.

Mots-clés : suivi, suivi multi-objets, analyse de fond, classification Bayesienne, histogrammes de couleurs, mouvements de caméra, points d'intérêt, Filtrage particulaire

1 Introduction

It is well known that global characterization of objects color distributions with histograms is a powerful means to track arbitrary non-rigid objects. Such a color modeling has been used in both deterministic algorithms (such as mean shift [1]) and probabilistic ones (such as particle filters [2, 3]). Anyhow, some tracking situations are still very challenging, especially when one or several of these events hold: presence of clutter, camera zooming, occlusions, appearance or illumination changes. In many applications, and in particular team sport analysis, another challenge arises when multiple objects with similar appearances (e.g., players of a given team) are jointly tracked.

The first main idea developed in this paper is that the robustness of such algorithms can be improved if the tracked entity is considered with respect to its background. We propose developments that rely on the analysis of the background in terms of color and motion. Firstly, we present a local color analysis of the surrounding background for improved discrimination, as well as for automatic diagnosis of occlusion and clutter situations, and selective adaptation. This is developed in section 2 in terms of Bayes classification of pixel colors, leading to an expression of the probability that a pixel belongs to an object as a function of its color. Secondly, in section 3, we address the robust dominant (camera) motion estimation, based on KLT point trackers spread over the image (following Kanade, Lucas and Tomasi, see references [4, 5, 6]). The pan-tilt-zoom parameters thus estimated are then used for conditioning the target dynamics used by tracking algorithms.

The second main topic of this paper is the extension to multi-object tracking. We propose in section 4 an efficient method (with few additional computation) in which, for K objects, K single trackers are firstly run as if there were no interactions between the targets, and then the results of each single object tracker are post-processed in order to incorporate the possible interactions. The aim is then to reduce the influence of pixels that are shared by several individual trackers. This sharing is driven by the probabilities of association between pixels and objects, which depend on the position and color of each pixel, and on the results of each individual tracker. This procedure can be applied whether the tracked entities share or not the same appearance and/or dynamical models.

Finally, some results are provided that demonstrate the efficiency of the overall tracking framework, with a special attention paid to team sports sequences.

2 Foreground and background color modeling

In tracking problems, rigid objects can appropriately be represented by a template, whereas non rigid objects are more suitably described by their color histograms [1, 3, 2].

Anyhow, the quality of a tracking procedure is partly led by the discrimination properties of the reference model of the entity in the context in which it appears. Comaniciu *et al.* [1] or Jaffré *et al.* [7] propose two different histogram manipulations, both inspired by color indexing developed in [8] for object recognition, to reduce the influence of background colors in tracking. We present hereafter a formalization of this approach, based on Bayes classification, that leads to a slightly different formulation. We show how this formalism can be

used to detect occlusions and clutter, and to handle appearance changes through selective adaptation.

2.1 Bayes classification of pixel colors

In tracking tools, a usual way to define a reference image consists in drawing a rectangle in a frame of the sequence in hand. Most of the time this operation also initializes the tracking (time and position). Let us assume that, through this operation, the user has roughly classified pixels into two classes: pixels that belong to the entity inside the selection (subset Ω_O), and pixels that do not, around the selection. This labeling must be thought as noisy, since, due to the simple shape of the selection, some pixels inside it actually belong to the background. Following the classical Bayes classification scheme [9], the probability that a pixel π with color components $\mathbf{I}_\pi$ (whatever the color space) belongs to an object is expressed as:

$$p\left(\pi \sim \mathcal{O} | \mathbf{I}_\pi \equiv u\right) = \frac{p\left(\mathbf{I}_\pi \equiv u | \pi \sim \mathcal{O}\right) p\left(\pi \sim \mathcal{O}\right)}{p\left(\mathbf{I}_\pi \equiv u\right)}, \tag{1}$$

where the color space is partitioned in B bins $b_u, u = 1 \dots B$. The notation $\mathbf{I}_\pi \equiv u$ denotes that the color $\mathbf{I}_\pi$ is in bin b_u of the color space. The learning procedure consists in estimating the three probabilities involved in (1) with the labeled examples and counter-examples provided by the user (i.e. the training set). If the size of this training set is large enough, these three amounts are directly estimated from color histograms. Let $H^O(u)$ be the non normalized histogram of the pixels in Ω_O:

$$H^O(u) = \sum_{\pi \in \Omega_O} \mathbf{1}_{b_u}[\mathbf{I}_\pi], \tag{2}$$

where $\mathbf{1}_{b_u}[.]$ is the indicator function of the part b_u. The corresponding normalized histogram is denoted as $h^O(u)$ and verifies $\sum_{u \in B} h^O(u) = 1$. Similarly, the histograms $H^T(u)$ and $h^T(u)$ of the whole training set are introduced. The probabilities involved in (1) can be approximated as follows:

$$p\left(\mathbf{I}_\pi \equiv u | \pi \sim \mathcal{O}\right) \approx p\left(\mathbf{I}_\pi \equiv u | \pi \in \Omega_O\right) \approx h^O(u) \tag{3}$$

$$p\left(\mathbf{I}_\pi \equiv u\right) \approx h^T(u). \tag{4}$$

The *a priori* probability that π belongs to the object $p(\pi \sim \mathcal{O})$ can be set to 0.5 (maximum likelihood approach) or computed from the size of both classes subsets (maximum *a posteriori* approach), if their proportions are supposed to be representative. In the latter case, the *a posteriori* distribution is approximated as

$$p\left(\pi \sim \mathcal{O} | \mathbf{I}_\pi \equiv u\right) \sim \frac{H^O(u)}{H^T(u)}. \tag{5}$$

This formulation can also be understood as the proportion of examples in the subset of pixels whose color is in b_u.

Unfortunately, the number of bins in the histograms, which should be sufficiently large to allow discrimination between colors, and the size of the selected object, which can be fairly small, hardly permit to assume that the database

is large enough. A color often appears neither in examples nor in counter-examples, which makes the probability in (5) undefined (by extension, a color that appears only a few times in the dataset may not be considered as discriminating). To overcome this difficulty, the probability in (5) can be reformulated by considering that it corresponds to the parameter ℓ of a binary random variable (the event $\pi \in \Omega_O$ is realized or not). The distribution of such a parameter, say $f(\ell|m,n)$, can be expressed as a function of the total number of simulations n and the number of positive simulations m ($m \leq n$). One has

$$f(\ell|m,n) = \frac{(n+1)!\ell^m(1-\ell)^{n-m}}{[m!(n-m)!]}. \tag{6}$$

The expectation of ℓ is then deduced from this distribution: $\mathbb{E}(\ell) = (m+1)/(n+2)$.

In our case, $n = H^T(u)$ and $m = H^O(u)$, hence:

$$p\left(\pi \sim \mathcal{O}|\mathbf{I}_\pi \equiv u\right) = \frac{H^O(u)+1}{H^T(u)+2}. \tag{7}$$

This formula is also valid when the number of pixels with color u in the training set is small. If no such pixels are present, the probability is 0.5. When lots of pixels with color u are present, the above expression tends asymptotically to the expression given by (5).

2.2 Background subtraction

From the above classification, a background subtraction can easily be developed. A new reference histogram H_0^* (and its corresponding normalized one, say h_0^*) is defined:

$$H_0^*(u) = \begin{cases} H^O(u) & \text{if } p\left(\pi \sim \mathcal{O}|\mathbf{I}_\pi \equiv u\right) \geq S \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where S is a threshold ($0 < S < 1$). By this means, colors that probably belong to background are cancelled from the reference model. Figure 2 gives an illustration of this method. The mask of the object pixels is given in white on black. Although the mask is a little noisy (this is because no spatial influence between colors is considered), the silhouette is correctly retrieved and the most specific colors are selected.

Comaniciu *et al.* [1] use histogram ratios to reduce the influence of background colors in the mean shift tracking algorithm, but in a less drastic manner: the denominator is the histogram of the neighborhood only (excluding the selection), and a threshold is required to avoid division by zero, so that a color is never completely cancelled. Contrary to our method, when this neighborhood is monochromatic (a situation that can appear in sports videos such as football, rugby, etc.) their subtraction is inefficient. Jaffré *et al.* [7] do not perform an explicit background subtraction, but apply a mean shift procedure on retroprojected histogram ratios, that correspond to $p\left(\pi \sim \mathcal{O}|\mathbf{I}_\pi \equiv u\right)$ approximated as in (5). In their method, the denominator is the histogram of the whole frame and is updated at each time step, thus adapting the background model. Anyhow they do not maximize the similarity between reference and candidate histograms, but the averaged probability that the candidate pixels belong to the object.

2.3 Occlusion and clutter diagnosis

The Bayes classification error is commonly used to evaluate the confidence on classification results. It corresponds to the probability of bad classification. We show in this section how this amount can be used to analyze the results of tracking algorithms and to detect occlusions or clutter.

When a tracking algorithm provides a new result, the output image region is a set of pixels Ω_t^O supposed to belong to the tracked object. Under this assumption, the classification error I_B is expressed as the probability averaged over all pixels in this region, that they are associated to the background (denoted as $\mathcal{B}$):

$$\begin{aligned} I_B &= \frac{1}{|\Omega_t^O|} \sum_{\pi \in \Omega_t^O} p\left(\pi \sim \mathcal{B} | \mathbf{I}_\pi\right) \\ &= \frac{1}{|\Omega_t^O|} \sum_{\pi \in \Omega_t^O} \left[1 - p\left(\pi \sim \mathcal{O} | \mathbf{I}_\pi\right)\right]. \end{aligned} \tag{9}$$

The output region is constrained to a given shape, so that some pixels in Ω_t^O have a color that more probably belongs to the background (under-optimal classification). Thus we also define a minimal Bayes error I_{min} obtained by letting each pixel in Ω_t^O being classified with the maximum *a posteriori* rule:

$$I_{min} = \frac{1}{|\Omega_t^O|} \sum_{\pi \in \Omega_t^O} \min\left[p\left(\pi \sim \mathcal{B} | \mathbf{I}_\pi\right); p\left(\pi \sim \mathcal{O} | \mathbf{I}_\pi\right)\right]. \tag{10}$$

We have $0 \leq I_{min} \leq I_B \leq 1$. When new colors appear (due for instance to illumination changes) the uncertainty of their classification is high and the corresponding probability to belong to the object is close to $1/2$. Both I_{min} and I_B will then increase. Conversely, when the tracked object is partially or totally occluded by a scene element whose colors have already been seen in the neighborhood, only I_B increases. Based on these remarks, we developed a method that aims at diagnosing occlusions and, contrary to other methods in literature [10, 11], at distinguishing between occlusions and appearance changes (this distinction is important to decide if an adaptation is opportune or not). This mechanism is based on the estimation of the proportion $\alpha_{\notin}$ of pixels in Ω_t^O that do not belong to the object. If $\min[p(\pi \sim \mathcal{B}|\mathbf{I}_\pi); p(\pi \sim \mathcal{O}|\mathbf{I}_\pi)]$ were constant in Ω_t^O, thus equating I_{min}, we would have:

$$\alpha_{\notin} = \frac{I_B - I_{min}}{1 - 2I_{min}}. \tag{11}$$

This formula is a convenient estimation even if the drastic underlying assumption does not fully hold, because the errors are averaged on the pixels (we only estimate a global behavior).

In the presence of appearance changes, the two errors increase but $\alpha_{\notin}$ is stable. Conversely, when occlusions arise, this amount increases. The detection of occlusions is thus done by detecting the variations in $\alpha_{\notin}$. An increase of 50% (respectively, 100%) means a partial (resp. total) occlusion. Clutter is detected similarly but with errors computed in the neighborhood Ω_t^B of the tracking result. Examples are given in figure 1.

Figure 1: Test sequence for occlusion diagnosis. The reference model is defined in the first image. Partial (images 3 and 5) and total (image 2) occlusions are correctly detected, as indicated by 'P' and 'O' letters, while illumination change (image 4) is not mistaken for occlusion. Here the adaptation process is applied as soon as no partial or total occlusion is detected, and a particle filter is used to track the hand.

2.4 Selective adaptation

In tracking algorithms, appearance changes lead to inaccurate or false results. It is thus tempting to update the reference model as new representations of the object are found. On the other hand, the risk of such a procedure is that it causes drift as background patterns or colors are introduced inside the model. To reduce this risk, we chose to apply adaptation only when no occlusion or partial occlusion is detected. Moreover, in many videos with controlled illumination and artificial lights, illumination changes are of secondary importance, while the main appearance modifications coincide with a positive camera zoom (because in this case, the object becomes larger, its histogram is richer and new colors appear). This is especially true for sports videos with fast and large zooms-in on specific game actions such as goals. For such applications, one can thus call the adaptation procedure only when the object size increases (see next paragraph for zoom estimation). A binary variable δ_t is set to 1 when adaptation is possible, 0 otherwise. The following adaptation procedure is performed at each time step:

$$H_t^O(u) = H_{t-1}^O(u) + \delta_t \sum_{\pi \in \Omega_t^O} \mathbf{1}_{b_u}[\mathbf{I}_\pi], u = 1 \dots B, \tag{12}$$

where H_t^O is the cumulated color histogram of the tracked object. Similarly, H_t^T is the cumulated histogram of the region including the object and its neighborhood. At $t = 0$, it corresponds to the initial histograms introduced in §2.1. These new histograms are then introduced in (7) to update the probability that a pixel belongs to the object, and the new reference histograms H_t^* and h_t^* are updated as in (8).

3 Background motion estimation

In sequences with important pan and tilt, the observed motion of an object is the combination of its real dynamics and the camera motion. Moreover, when zooms are present, object size variations occur that can make tracking algorithms fail or loose accuracy. Methods that determine the size directly from the image data, such as in Zivkovic *et al.* [12], would be sensitive to clutter or occlusions. Similarly, methods based on particles would require a large amount of particles in order to sample a wide range of possible sizes and positions. Therefore, it can be useful to calculate the camera (dominant) motion, and compensate for it in the object motion, thus reducing the searched domain of possible target states.

Estimation of dominant apparent motion has been addressed in the literature by different means. Sparse optical flows [13, 14] or global image data [15, 16] can be used. A robust estimation of a parametric model is usually applied in order to deal with outlier motions of objects moving in the scene. The method we present aims at combining the advantages of the different existing approaches. Sparse fields obtained from pyramidal KLT computation [6] are preferred so that attention is focused on interest points where the computed flow has the highest confidence [4]. The robust estimation is performed with iterated weighted least squares, preferred to a RANSAC procedure so as to obtain a reproducible result and to manage the computation time more easily. We also derive the variance of the parameters and propose a sequential filtering

to deal with abrupt illumination changes (e.g. flashes). Moreover, the model has only three parameters since the only motions to be considered in our applications are pan, tilt and zoom (homographies, as in [13, 14], involve 8 parameters, which are too many degrees of freedom for our purpose). At last, our algorithm does not depend on a model description of the ground as in [13, 14].

3.1 Robust affine estimation of dominant motions

Local motion vectors, say $\mathbf{w}_i = (u_i, v_i)^T, i = 1 \dots N$, are computed with the pyramidal KLT method described in [6], at N interest points denoted $(x_i, y_i)^T$. We aim at estimating the parameters $\boldsymbol{\theta}$ (the horizontal translation θ_1, the zoom θ_2 and the vertical translation θ_3) of the global affine model expressed as

$$\hat{\mathbf{w}}_i = \begin{pmatrix} \hat{u}_i \\ \hat{v}_i \end{pmatrix} = X_i \boldsymbol{\theta} = \begin{bmatrix} 1 & x_i & 0 \\ 0 & y_i & 1 \end{bmatrix} \boldsymbol{\theta}, \tag{13}$$

where the hats represent estimated quantities. We suppose that a major part of the KLT features are located on the background and move as a function of the camera motion. The following robust estimation method, based on M-estimators and iterated weighted least squares technique [17], aims at suppressing the influence of secondary motions, due to any moving object in the scene.

One defines the residuals at each feature as $r_i = \|\mathbf{w}_i - \hat{\mathbf{w}}_i\|$. The estimation is led by minimizing the amount $\sum_{i=1}^{N} \rho_M(r_i)$, where ρ_M is the so-called M-estimator [17]: it replaces the square function that is used in the classical (but non-robust) least squares technique, and aims at reducing the influence of large residuals due to outlier features. Here the M-estimator used is the Tuckey biweight function with parameter C. It can be shown that the parameters are estimated by solving the following equation, where the weight ω_i determines the influence of the feature i:

$$\sum_{i=1}^{N} \omega_i r_i \frac{\partial r_i}{\partial \theta_k} = 0. \tag{14}$$

The weights are defined by:

$$\omega_i = \frac{1}{r_i} \frac{\partial \rho_M}{\partial r}(r_i). \tag{15}$$

Once the weights have been fixed, the resolution of (14) can be done by a least squares estimation. After some algebraic manipulations, the estimated parameters are expressed by

$$\hat{\boldsymbol{\theta}} = A^{-1}(X^{(1)T} W \mathbf{u} + X^{(2)T} W \mathbf{v}), \tag{16}$$

with $A = X^{(1)T} W X^{(1)} + X^{(2)T} W X^{(2)}$, and where $X^{(1)}$ and $X^{(2)}$ are the $N \times 3$ matrices with i-th rows equal to $(1, x_i, 0)$ and $(0, y_i, 1)$, respectively. The matrix W is the weight diagonal matrix whose i-th diagonal term is ω_i. The N-dimensional vectors $\mathbf{u}$ an $\mathbf{v}$ are the horizontal and vertical components of the motion at all features.

The idea behind the iterated least squares method is to start with unweighted features, to perform a first estimation and to update the weights as a function of the residuals found at each feature for this estimation (this leads to a modification of the M-estimator parameter). The procedure goes on iteratively with a

new estimation, new weights and so on. In practice, the first estimation is here performed with all $\omega_i = 1$, then the residuals and $C = 4\text{med}_i|r_i|$ are computed, and the weights are updated as $\omega_i = (r_i^2 - C^2)^2$ if $|r_i| \leq C$, 0 otherwise (corresponding to the Tukey function). The process is iterated a few times (four in our experiments). All features that end with null weights correspond to outliers.

Figure 2 shows the inliers and outliers found in a sequence with a man walking behind trees. The features on the moving objects (man, car) or with corrupted local motion are detected as outliers, whereas the features in the background are inliers. The mosaic representation built from the estimated camera motion gives a visual validation of the method.

Figure 2: First image: background subtraction performed with the Bayes classification (the pixels selected are inside the orange rectangle). Images 2, 3 and 4: Background motion estimation. The red plots are inlier features and the blue ones are outliers, the lines are the corresponding flows. Bottom: Mosaic image reconstructed with the estimated camera motion.

3.2 Uncertainty and sequential filtering

Estimating uncertainties in robust estimation is not a trivial task, and often requires some approximations [18]. We derive a formula in which the weights are assumed to be non random variables. The optical flows are written as a

function of their estimates, as $u_i = \hat{u}_i + \xi_{ui}$ and $v_i = \hat{v}_i + \xi_{vi}$, where ξ_u and ξ_v are random variables with variances σ_u^2 and σ_v^2 respectively and covariance σ_{uv}, which can be evaluated as the empirical variance matrix of the residuals. These expressions can be introduce in (16) and, since $\hat{\mathbf{u}} = X^{(1)}\boldsymbol{\theta}$ and $\hat{\mathbf{v}} = X^{(2)}\boldsymbol{\theta}$, we have:

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\theta} + A^{-1}(X^{(1)T}\xi_u + X^{(2)T}\xi_v) \tag{17}$$

The uncertainty of the parameters is directly derived from this formula:

$$\begin{aligned} \mathrm{var}\hat{\boldsymbol{\theta}} = \quad & A^{-1}\left[\sigma_u^2 X^{(1)T} W X^{(1)} + \sigma_v^2 X^{(2)T} W^2 X^{(2)}\right. \\ & \left. +\sigma_{uv}\left(X^{(1)T} W^2 X^{(2)} + X^{(2)T} W^2 X^{(1)}\right)\right] A^{-1}. \end{aligned} \tag{18}$$

The estimated parameters and their variance can be thought as a Gaussian measurement model in a hidden Markov chain, in which the hidden state is the true parameters vector. If we assume that this state vector is driven by a Gaussian dynamics of order one (this corresponds to the assumption that changes in the camera motion are smooth), a sequential Kalman filtering is made possible. The Gaussian dynamics hypothesis has been checked on a set of sequences where ground truth camera motions were available, and the covariance matrix has been evaluated experimentally. The *a posteriori* estimate given by the Kalman filter is denoted as $\hat{\boldsymbol{\theta}}_t^K$.

This method is illustrated in figure 3, on a sequence that contains camera flashes. For instance, the KLT method fails at frame #16, which leads to a false affine estimation with a large uncertainty. The Kalman filter allows to recover a correct result, so that we can build the stroboscopic view of the sequence.

4 Tracking algorithm

In this section we present a tracking algorithm developed for the following context: non-rigid targets, presence of clutter, occlusion events, size changes and camera motions, appearance and illumination changes, and multiple objects with similar appearance. The non-rigid characteristics of the object are suitably taken into account through a color histogram representation, which provides a global description. Within a probabilistic Bayesian framework, a particle filter is used for its capabilities to deal with occlusions and multimodal solutions, as for instance when the localization of the target is ambiguous for a while due to clutter or partial/total occlusion. Another part of the difficulties mentioned hereabove are tackled by the developments presented in the previous sections: background color modeling (for a better description of the object and the account of appearance changes through an adaptation procedure) and background motion (for large camera motion and zoom) are introduced in the particle filter. An extension to multiple targets, which is valid whether tracked objects are identical or simply similar, has also been developed and is described hereafter. The section 4.1 describes the robust tracking algorithm valid for one object. The section 4.2 is devoted to multi-object tracking: methods from the literature are discussed and the new one is described.

4.1 A particle filter with background considerations

The underlying question in a single-object tracking problem is to evaluate the state of the target (at least its position) at each time step as a function of the

Figure 3: Ball tracking in the presence of flashes. First four images: Images #1 and 16 (top), #34 and 55 (bottom); local KLT motions plotted at interest points, in red (inliers) and blue (outliers); Middle: stroboscopic view of ball tracking results on image 60; the tracking results (every 5 time steps) are overlaid on the scene after compensation of the filtered estimate of camera motion. Bottom: estimated camera motion parameter $\hat{\theta}_2$ (green) and the corresponding filtered one $\hat{\theta}_2^K$ (red).

data provided (i.e. the pixel colors). The state is related to the data through a measurement model, which is, in our case, highly non linear and possibly corrupted by an arbitrary noise. Also, the nature of the target leads to a temporal coherence between consecutive positions. This consistency is captured by the target's dynamics, or system model. These two models form a sequential Markov chain and Bayesian filtering is used to solve the tracking problem [19].

In what follows, a hidden state at time t, say $\mathbf{e}_t$, is defined as the concatenation of the object's position $\mathbf{x} = (x_t, y_t)^T$, its size s_t (area) and its two velocity components u_t and v_t. The first three components of this state vector define a subset of pixels in the image, denoted as $\Omega^O(\mathbf{e}_t)$, that is a candidate region potentially occupied by the tracked object (the ratio between the object width and height is here assumed constant).

In a color-based particle filter (see [2, 3] for details), weighted particles are used to sample the filtering distribution of states $p(\mathbf{e}_t|\mathbf{z}_{1:t})$, where $\mathbf{z}_{1:t}$ denotes the concatenation of the data frames 1 to t. The set of M particles at time t is denoted as $\{\mathbf{e}_t^{(i)}\}, i = 1 \dots M$, and the corresponding weights as $w_t^{(i)}$. Particle filter consists in the following iterative procedure (we use the so-called *Bootstrap* version [2]):

- **Prediction step:** From the current set of particles $\{\mathbf{e}_{t-1}^{(i)}\}$, propagate them up to time t by sampling from the dynamics.
- **Filtering step:** Compare the predicted particle to the measurement, through the likelihood function $p(\mathbf{z}_t|\mathbf{e}_t^{(i)})$. This gives updated weights defined by: $w_t^{(i)} \propto w_{t-1}^{(i)} p(\mathbf{z}_t|\mathbf{e}_t^{(i)})$ with $\sum_i w_t^{(i)} = 1$.
- **Importance Resampling step:** (optional) Draw M samples from the distribution described by $\{\mathbf{e}_t^{(i)}, w_t^{(i)}\}$ so as to have a new set of particles with equal weights.

This iterative procedure can be initialized either automatically (e.g. with a color blob detector [2], or an object detector [20]) or manually. In the latter case, a location is provided at an initial time, and a Gaussian distribution around this state is assumed to generate the initial particle set.

We will now define the likelihood function and the dynamics.

In order to evaluate the agreement between the data and a candidate state vector $\mathbf{e}_t$ (i.e. the likelihood), the whole image is divided into two sets of pixels: those inside the candidate region, $\Omega^O(\mathbf{e}_t)$, and the others lying in $\overline{\Omega^O}(\mathbf{e}_t)$. If the colors of the pixels in these two sets are assumed independent, and if the object is defined by its reference histogram h_t^* and the background by a histogram b^*, the likelihood can then be written as

$$p(\mathbf{z}_t|\mathbf{e}_t) = p\left(h(\Omega^O(\mathbf{e}_t)) \sim h_t^*\right) p\left(h(\overline{\Omega^O}(\mathbf{e}_t)) \sim b^*\right), \tag{19}$$

where $h(\Omega)$ is the color histogram of a set of pixels Ω, and where the expression $h(\Omega) \sim h^*$ reads *the colors in the set of pixels* Ω *are generated by the model* h^*. It is usual to neglect the influence of the second term $p\left(h(\overline{\Omega^O}(\mathbf{e}_t)) \sim b^*\right)$ in the expression of the likelihood (19), either because the background model is not known precisely or because the tracked object is small compared to the image, i.e. $|\Omega^O(\mathbf{e}_t)| \ll |\overline{\Omega^O}(\mathbf{e}_t)|$, so that the histogram of the subset $\overline{\Omega^O}(\mathbf{e}_t)$ can be

approximated by the one of the whole image, which does not depend upon $\mathbf{e}_t$. This approximation also has a practical advantage: it prevents from computing large image histograms.

One defines:

$$\begin{aligned} p(\mathbf{z}_t|\mathbf{e}_t) &\simeq p\left(h\left(\Omega^O(\mathbf{e}_t)\right) \sim h_t^*\right) \\ &\propto e^{-\lambda\left[1-\rho\left(h_t^*, h\left(\Omega^O(\mathbf{e}_t)\right)\right)\right]}, \end{aligned} \tag{20}$$

where λ is an empirical parameter (set to 20 in our experiments) and ρ is the Bhattacharyya similarity coefficient, which compares two histograms. This amount is defined by a sum over the histogram bins as:

$$\rho\left(h_t^*, h\left(\Omega^O(\mathbf{e}_t)\right)\right) = \sum_{u=1}^{B} \sqrt{h_t^*(u), h\left(\Omega^O(\mathbf{e}_t), u\right)}. \tag{21}$$

Background subtraction, possibly with adaptation, can be introduced through the definition of h_t^* given in (8) and (12).

The background camera motion, estimated as explained in the above section, is introduced into the dynamics of the particle filter, so that the remaining random part is directly correlated to the object's dynamics in the real scene.

This conditioning of the dynamical model on estimated camera motion improves the robustness and accuracy of the tracker. Indeed, the random part of the dynamics, which drives the evolution of the set of particles, does not need to include uncertainty due to an unknown camera motion. Choosing a first order autoregressive dynamics, we set:

$$\mathbf{e}_t = D_t \mathbf{e}_{t-1} + \mathbf{b}_t + \boldsymbol{\nu}_t, \tag{22}$$

where $\boldsymbol{\nu}_t$ is an additive Gaussian noise describing the possible motions of the object relatively to the scene, and with $\mathbf{b}_t = (\hat{\theta}_{t1}^K, \hat{\theta}_{t3}^K, 0, 0, 0)^T$. The matrix D_t is expressed by:

$$D_t = \begin{bmatrix} \alpha_t & 0 & 0 & \alpha_t & 0 \\ 0 & \alpha_t & 0 & 0 & \alpha_t \\ 0 & 0 & \alpha_t^2 & 0 & 0 \\ 0 & 0 & 0 & \alpha_t & 0 \\ 0 & 0 & 0 & 0 & \alpha_t \end{bmatrix}, \tag{23}$$

where $\alpha_t = 1 + \hat{\theta}_{t2}^K$.

Since the behavior of the particles is described by parameters that are estimated on the image (through the procedure introduced in section 3), the filters used must be thought as conditional filters. It has been proved that such filters have the same properties as Bayesian filters derived for standard state space models [21].

Remark

The knowledge of the background motion can also be used to improve the capabilities of a kernel-based tracking algorithm, such as the mean shift procedure proposed by Comaniciu *et al.* [1]. The mean shift procedure (see [1] for details) aims at finding a local maximum of the Bhattacharyya coefficient, starting the search at the previously estimated position $\hat{\mathbf{x}}_{t-1}$. When the camera affine model

is known, this initial position can be modified accordingly as $\alpha_t \hat{\mathbf{x}}_{t-1} + (\hat{\theta}^K_{t1}, \hat{\theta}^K_{t3})^T$, with $\alpha_t = 1 + \hat{\theta}^K_{t2}$. The latter coefficient is also used to update the size of the kernel function used in the procedure, which represents the size of the object. The first experiment in section 5 illustrates the interest of this compensation. In particular, it is compared to another method, proposed by Zivkovic *et al.* [12], that aims at tracking size variations, with a procedure closely related to Comaniciu *et al.* 's mean shift.

4.2 Extension to multi-object tracking

When tracking jointly the positions in the image frame of several similar objects (as opposed to track their positions in the 3D world, e.g., players' positions on the field), problems arise when some of these objects get visually close or, worse, when they partially or totally occlude each other. For that reason, running jointly independent single object trackers will not work in general. An appealing solution consists first in defining a joint tracker in the product multi-object state space and second in introducing some kind of "exclusion principle" within the observation model. This principle, which prevents a piece of visual measurements from being simultaneously associated to different tracked entities, amounts to introduce depth ordering variables for objects occupying the same image portion (according to their current estimated filtering distributions). In the context of particle filtering, these nuisance variables can then be marginalized out, either by sampling [22] or by enumeration leading to a mixture observation model [2]. In both cases, the resulting joint particle filter becomes unpractical for more than few objects.

Besides this interaction of the co-existing individual states via the observation model, note that interactions via the dynamics can also be introduced, to capture for instance group behaviors (different players aiming at the ball). However, such dynamical interactions have been mostly proposed to enforce physical exclusion when tracking real-world positions of solid objects. In all cases, a substantial complexity is added, requiring for instance the use of internal MCMC iterations at each time step [23].

In a completely different perspective, Vermaak *et al.* [24], followed by Okuma *et al.* [20], propose a method that makes possible to treat explicitly, within the same particle filter, the multimodality induced by multiple objects with similar appearance. The aim is to cluster the particles such that each cluster captures one mode of the filtering distribution, and maintains it through time (in a traditional particle filter, the resampling induces a fast disappearance of the secondary modes to the profit of the principal mode).

Although this approach might be used to track jointly multiple objects (provided that they share the same appearance model and dynamics), it is not a proper multiple object tracking technique: it can't be extended to multiple objects with different or changing appearance/dynamical models; and there is a single tracker shared by the different targets. Also, no matter how many objects are tracked, the marginal filtering distribution associated to each individual target can be significantly multimodal, due for instance to clutter or other scene changes that are not due to tracked objects, such as partial occlusions.

By contrast with previous approach, the method presented here is truly multi-object *and* multimodal, while avoiding the complexity of methods based on joint sequential importance sampling in product state space. Moreover, it

neither requires a data association step as in Cai *et al.* [25] or Li *et al.* [26], nor the use of future informations (batch detection over the whole sequence) as in Pitié *et al.* [27] or Nillius *et al.* [28].

Let $\mathcal{O}_k, k = 1 \dots K$ denote the K objects to track, and $\mathbf{e}_{k,t}$ be the corresponding state vectors at time t.

In the single object particle filter, it is implicitly assumed, through the expression of the likelihood (20), that if a region has high similarity with the target, the pixels are necessarily associated to this target. A common failure of this approach is that, when two similar objects are tracked with two single trackers, these two trackers often lock on the same object after they have crossed each other if no form of exclusion principle is introduced.

To overcome this problem as well as the limitations of methods based on an exclusion principle (as discussed earlier), we redefine for each object the following likelihood:

$$p(\mathbf{z}_t|\mathbf{e}_{k,t}) \simeq p\left(h\left(\Omega^O(\mathbf{e}_{k,t})\right) \sim h^*_{k,t}\right) p\left(\Omega^O(\mathbf{e}_{k,t}) \sim \mathcal{O}_k\right), \tag{24}$$

that can be used within a multiple object tracker. Here $p\left(\Omega^O(\mathbf{e}_{k,t}) \sim \mathcal{O}_k\right)$ is the probability that the subset of pixels $\Omega^O(\mathbf{e}_{k,t})$ is associated to the k-th object. This quantity will be calculated from the individual probability of each pixel in the set, as a function of their position and color. Let π be a pixel in the image, $\mathbf{x}_\pi$ its coordinates and $\mathbf{I}_\pi$ its color values. One wants two evaluate the probability that π is associated to object k, say $p\left(\pi \sim \mathcal{O}_k|\mathbf{x}_\pi, \mathbf{I}_\pi\right)$. The Bayes theorem yields

$$p\left(\pi \sim \mathcal{O}_k|\mathbf{x}_\pi, \mathbf{I}_\pi\right) = \frac{p\left(\mathbf{x}_\pi|\pi \sim \mathcal{O}_k, \mathbf{I}_\pi\right) p\left(\pi \sim \mathcal{O}_k|\mathbf{I}_\pi\right)}{\sum_{\ell=1}^{K} p\left(\mathbf{x}_\pi|\pi \sim \mathcal{O}_\ell, \mathbf{I}_\pi\right) p\left(\pi \sim \mathcal{O}_\ell|\mathbf{I}_\pi\right)} \tag{25}$$

The quantity $p\left(\mathbf{x}_\pi|\pi \sim \mathcal{O}_k, \mathbf{I}_\pi\right)$ in equation 25 is closely related to the filtered distribution $p_k\left(\mathbf{e}_{k,t}|\mathbf{z}_t\right)$ obtained from a single object tracker, since the expression $\pi \sim \mathcal{O}_k$ denotes that no confusion with other objects can be made as is implicitly done in a single object particle filter. Also, the variable $\mathbf{I}_\pi$ is an element in the set of measures $\mathbf{z}_t$, so that the only difference between p_k and $p\left(\mathbf{x}_\pi|\pi \sim \mathcal{O}_k, \mathbf{I}_\pi\right)$ is that it focuses on individual pixels instead of state vectors. So, if p_k is represented by a set of M_k weighted particles $\{(\mathbf{e}^{(i)}_{k,t}, w^{(i)}_{k,t})\}, i = 1 \dots M_k$, each pixel is assigned the sum of weights of particles that contain it, that is:

$$p\left(\mathbf{x}_\pi|\pi \sim \mathcal{O}_k, \mathbf{I}_\pi\right) = \sum_{i=1}^{M_k} w^{(i)}_{k,t} \mathbf{1}_{\Omega^O(\mathbf{e}^{(i)}_{k,t})}(\mathbf{x}_\pi), \tag{26}$$

where $\mathbf{1}_\Omega(\mathbf{x})$ is the indicator function equaling 1 if $\mathbf{x}$ belongs to the image region Ω, and 0 otherwise.

The amount $p\left(\pi \sim \mathcal{O}_k|\mathbf{I}_\pi\right)$ in equation 25 is the probability that a pixel belongs to an object given its color. This probability is discussed in paragraph 2.1 and is given by (7).[1] We introduce the notation $\beta_k(\pi)$, defined for one pixel π and one object k, as follows (the dependence on t has been dropped):

$$\beta_k(\pi) = p\left(\mathbf{x}_\pi|\pi \sim \mathcal{O}_k, \mathbf{I}_\pi\right) p\left(\pi \sim \mathcal{O}_k|\mathbf{I}_\pi\right). \tag{27}$$

[1] Note that a more rigorous formulation could be developed for multiple objects if a classification with $K+1$ classes were used instead of 2 as in paragraph 2.1. Anyhow, it would have no influence on our multi-object algorithm.

If $\beta(\pi) = \sum_\ell \beta_\ell(\pi)$, the probability that a pixel π is associated to the k-th object is expressed by $p\left(\pi \sim \mathcal{O}_k | \mathbf{x}_\pi, \mathbf{I}_\pi\right) = \beta_k(\pi)/\beta(\pi)$. In the expression of the likelihood (24), the probability involved concerns a set of pixels instead of just one pixel. Since these pixels are not independent, this amount cannot be considered as the product of the individual probability of each pixels. Indeed, the probability of the ensemble would equal zero as soon as one pixel does not have a color in the reference histogram of the object, which is too drastic because the rectangular regions described by the state space only roughly fit the actual shape of the object. Thus, we define the following expression for the global probability of association:

$$p\left(\Omega^O(\mathbf{e}_{k,t}) \sim \mathcal{O}_k\right) = \frac{\sum\limits_{\pi \in \Omega^O(\mathbf{e}_{k,t}^{(i)})} \beta_k(\pi)}{\sum\limits_{\pi \in \Omega^O(\mathbf{e}_{k,t}^{(i)})} \beta(\pi)}. \tag{28}$$

The procedure for multi-object tracking is summarized in algorithm 4.2. The single object tracking procedure is repeated for each object, then the weights of the particles are modified in order to take into account the vicinity of other objects. In this procedure, it is neither required that the objects have the same reference histogram (targets can have only a few common colors) nor the same dynamics or number of associated particles. The individual particle filters can also have their own adaptation procedure so that in an application such as team players tracking, the initial reference histogram is common (the colors of the team, that can be introduced in a color based detector) but some differences (e.g. hair color) are learnt with the adaptation procedure. Note also that in the case of total occlusion of one tracked object by another similar tracked object, the two associated particle sets are allowed to be similar, as they should. Indeed, the particle weights are modified by the same factor in equation 28 so that after normalizing them in each set, they remain unchanged. By contrast, in the same situation, a method based on clustering [24] would lead to an identity loss (clusters are merged) and a method that uses a visibility variable, as in [29], would cause the two particle sets to behave differently, with a tighter cloud for the occluding object and widely spread cloud for the occluded object.

5 Results

One key feature of the proposed tracking framework is its ability to deal with size variations, providing they are induced by camera zooms. The first two situations treated below involve such camera motions. In figure 4 we present comparisons between different approaches based on mean shift. The advantage of using both background subtraction and camera motion compensation is clearly underlined since the corresponding tracker (in red) is the only one that tracks correctly the person from beginning to end, even when there are important zoom and pan. Without color background subtraction, the tracker is distracted by a tree, whose color falls in the initial selection. The tracker developed by Zivkovic *et al.* [12] is also distracted by clutter (tree with color similar to the man's coat). Without camera motion compensation, the mean shift tracker is able neither to take the size changes into account, nor to track the man correctly when the large pan occurs (last images).

Algorithm 1 Probabilistic multi-object tracking

Data :
K weighted particle sets $\{\left(\mathbf{e}_{k,t-1}^{(i)}, w_{k,t-1}^{(i)}\right)\}$,
the t-th frame of the sequence.
Results :
K weighted particle sets $\{\left(\mathbf{e}_{k,t}^{(i)}, w_{k,t}^{(i)}\right)\}$.
for all $k = 1 \dots K$ **do**
  Obtain new set of particles $\{\left(\mathbf{e}_{k,t}^{(i)}, w_{k,t}^{(i)}\right)\}$ with the single object particle filter described in part 4.1
  Compute $\beta_k(\pi)$ for all pixels in the image, with eqs. (27), (26) and (7).
end for
Compute $\beta(\pi) = \sum_{k=1}^{K} \beta_k(\pi)$ for all pixels in the image
for all $k = 1 \dots K$ **do**
  for all particles $i = 1 \dots M_k$ **do**
    Update weights with eq. (28) :
    $w_{k,t}^{(i)} \leftarrow w_{k,t}^{(i)} p\left(\Omega^O(\mathbf{e}_{k,t}) \sim \mathcal{O}_k\right)$
  end for
  Normalize weights, so that $\sum_{i=1}^{M_k} w_{k,t}^{(i)} = 1$.
end for

The second example concerns a video with occlusions during a zoom in a sport video extracted from the CVBASE'06 dataset.[2] The particle filter is used with camera motion compensation and background subtraction. Since the estimation of the zoom is performed on the whole image, the size of the tracked player can be updated even during the occlusion by the other player (results in blue). This figure also aims at showing how important can be the adaptation procedure in some situations. Here, the green player moves from one part of the field where the light is low to another one where the light is brighter. His appearance thus changes during the sequence and, moreover, the dark grey color of the other player is close to his dark green color at the beginning of the sequence. We can see that without adaptation (results in magenta), the light green color that appears in the second image is an unknown color and the likelihood in the region of the green player is low. The grey player has some common colors inside the reference histogram obtained from the first image. As a consequence, some particles are located on him. This is the reason why the tracking result (which is the average of all the particles) is between the two players in the second and fourth image. When the adaptation is on, the light green is learnt as belonging to the tracked player and the grey as belonging also to the background (the other player), which leads to the correct result.

The last three figures 6, 7 and 8 demonstrate the capabilities of the multi-object approach developed in this paper. This football sequence is from the SCEPTRE dataset[3]. Figure 6 shows how the kernel-based approach (here Comaniciu *et al.* 's approach is used [1] together with a background subtraction on the reference histogram) fails in tracking the white players. This deterministic

[2] http://vision.fe.uni-lj.si/cvbase06/downloads.html
[3] http://sceptre.king.ac.uk/sceptre/default.html

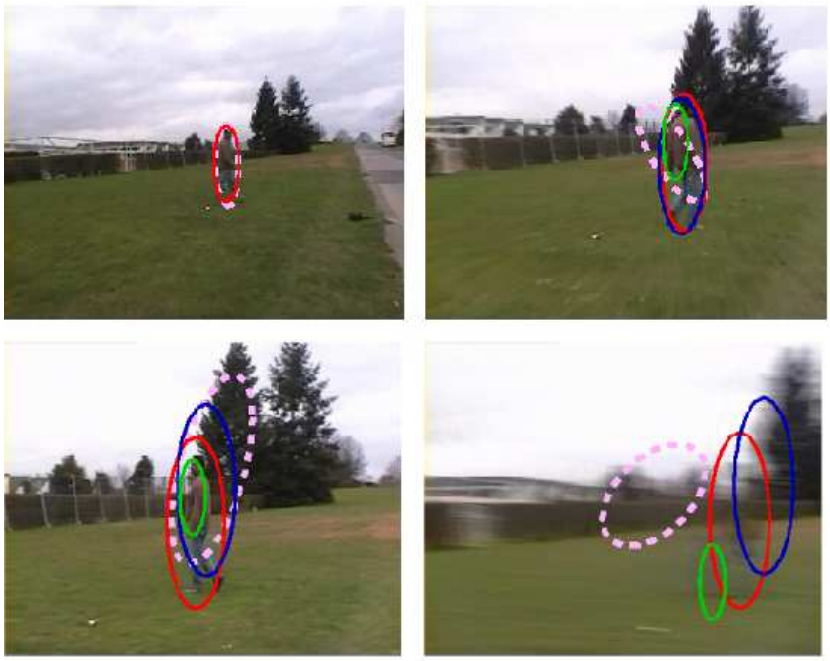

Figure 4: Comparison between mean shift tracking methods on a sequence with important camera motions (images #1, 25, 105 and 114). In red: with camera motion compensation and background subtraction. In blue: with camera motion compensation only. In green: with background subtraction only. In pink: results of [12].

method finds the closest maximum of the likelihood function, that is not the proper one as soon as similar objects cross each other. Moreover, since the procedure is initialized with the result obtained at the previous time step, the error is unrecoverable. Independent individual particle filters, whose results are given in figure 7, have a different behavior. Both trackers capture bimodal distributions when the players are close to one another (for each tracker, some particles lay on the two players). The multimodality is maintained for a while but the average particle is sometimes on one player and sometimes on the other one. The two individual trackers have very similar reference histograms, and their resulting average particles are almost always on the same player. At the end of the sequence, the multimodality is lost, and all the particles of the two trackers are on the same player. The multi-object procedure (see figure 8) reduces the influence of particles that interact with the other tracker, so that the tracked distributions correctly split after the players have crossed. One can see that two players are correctly tracked all along the sequence. Let us notice that a number of different runs have been performed on this example: the algorithm always succeeded in separating the two players after the crossing, but sometimes swapped them; the two players look so similar in the two images that there is no way of distinguishing them, especially with colors only.

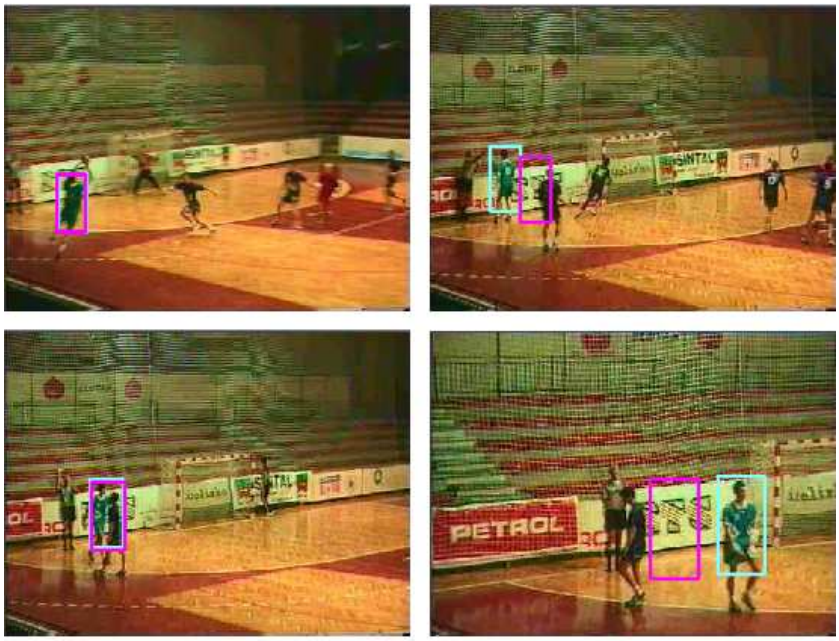

Figure 5: Probabilistic tracking of size changes during occlusion in clutter. Handball sequence extracted from the *HandballC.avi* file of the CVBASE'06 dataset, images #10828, 10883, 10948 and 10986. A particle filter is used with camera motion compensation and with (in blue) or without (in magenta) adaptation. When the adaptation is on, it is performed as soon as no partial or total occlusion is detected.

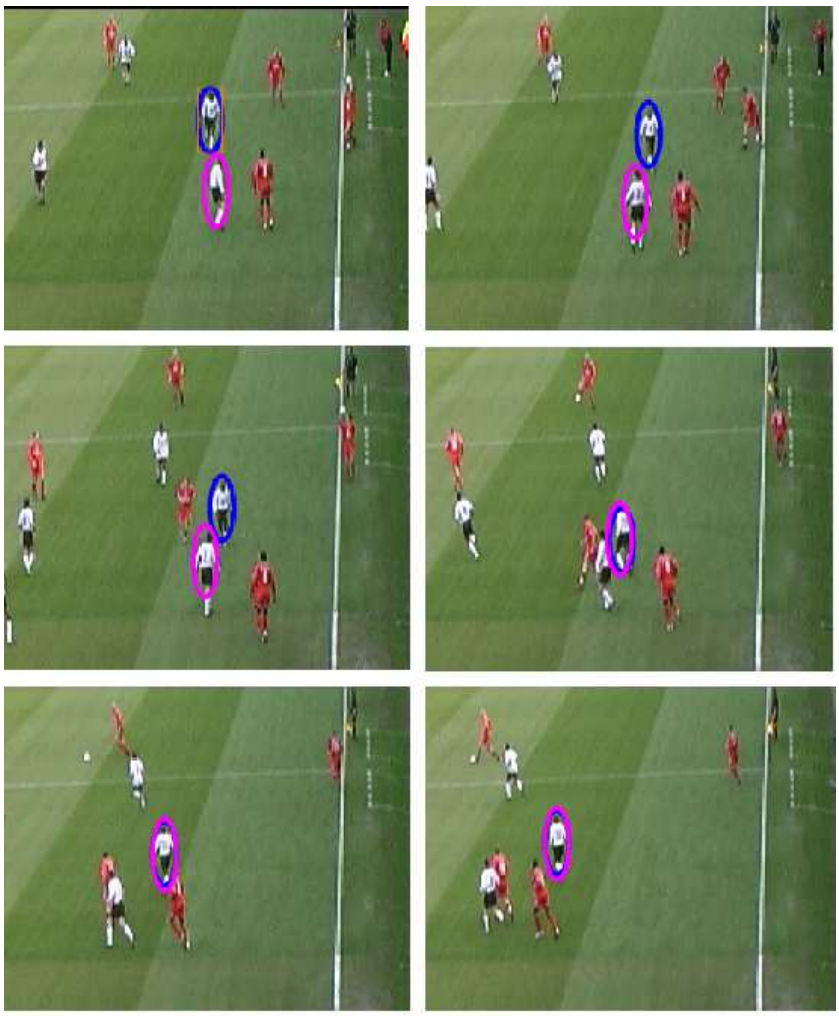

Figure 6: Football players tracking with the kernel based approach. Images #1, 45, 135, 190, 220 and 240. Two trackers (blue and magenta) are initialized in the first image on two different players with similar appearance.

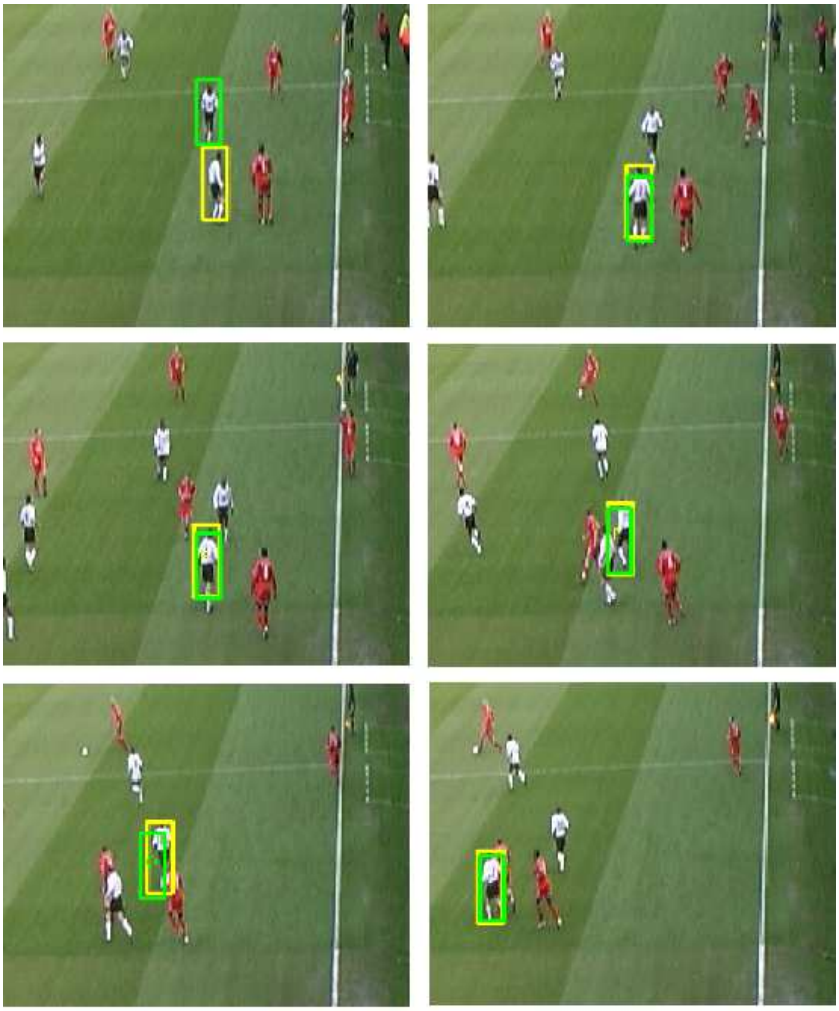

Figure 7: Football players tracking with two independent individual particle filters. Same images and initializations as in figure 6.

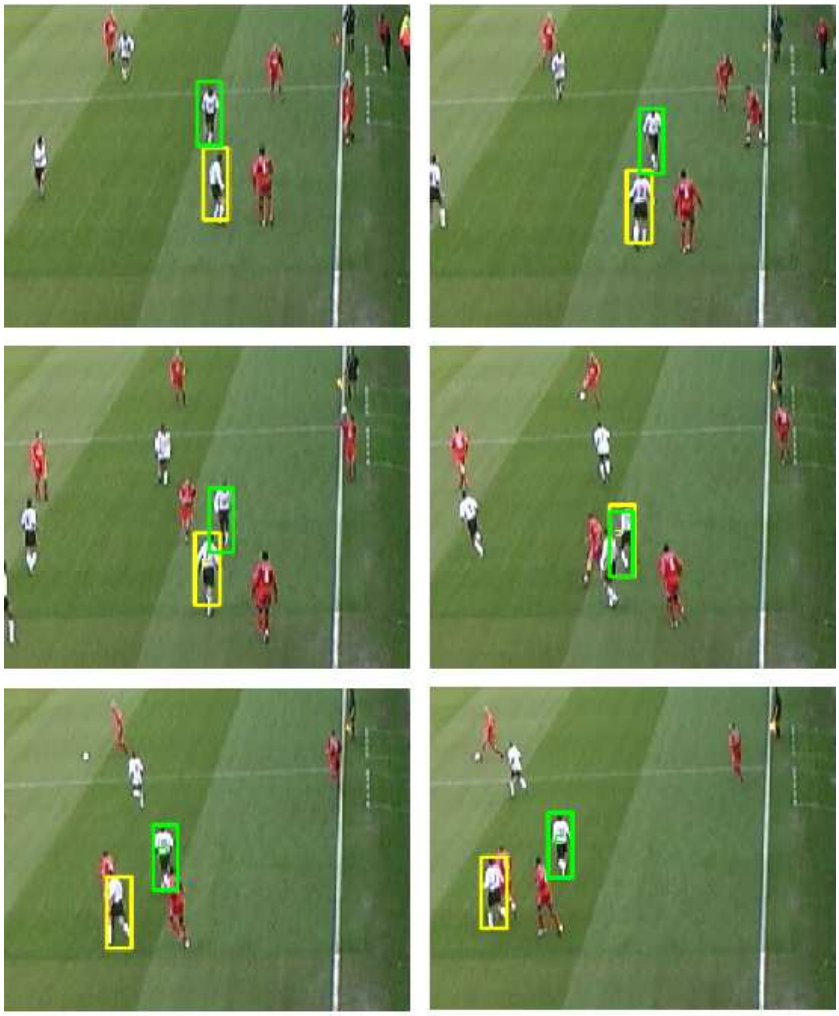

Figure 8: Football players tracking with our multiple object algorithm. Same images and initializations as in figure 6.

6 Conclusions

In this paper we have shown that probabilistic color based tracking can be improved with background analysis. More specifically, subtraction of background colors in the reference histogram as well as compensation of the background motion (due to camera pan, tilt and zoom) into the dynamics of the tracked object have been developed. In addition, an efficient extension has also been proposed in order to deal with multiple objects, possibly similar and close to each others.

The merit of these developments has been demonstrated on sequences involving large camera motions, zooms, occlusions, clutter, appearance changes, and multiple identical objects. They have been applied in particular to players tracking in team sports videos, in which context the distinctive features of our approach are especially valuable.

A first perspective concerns the handling of varying numbers of objects of interest in the viewfield within the multiple object tracker. Although not mentioned in this paper, this problem has already received some attention in the literature, and different techniques have been proposed to generate "births" and "deaths" of objects in a particle filter framework (including birth processes based on color-blob detectors [30] or application-dependent object detectors [20]). Such techniques could easily be used in the multi-object tracking approach that we propose. In the context of player tracking, this would allow the initialization of a new individual filter for each player of a given team that enters the image, and the elimination of the individual filter associated to a player exiting the image.

On a more prospective side, the introduction of application-dependent high-level dependencies between the different individual trackers (e.g. based on the rules or tactic of a given sport), which can be designed in an ad-hoc way or learnt from labeled data, remains an interesting and open problem.

7 Acknowledgements

This work has been supported by France Telecom R&D.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–575, 2003.

[2] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Proc. Europ. Conf. Computer Vision*, 2002.

[3] K. Nummiaro, E. Koller-Meier, and L. Van Gool. A color-based particle filter. In *ECCV Workshop on Generative-Model-Based Vision*, 2002.

[4] J. Shi and C. Tomasi. Good features to track. In *Proc. Conf. Comp. Vision Pattern Rec.*, June 1994.

[5] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In P. J. Hayes, editor, *Proc. 7th Int. Joint Conf. Artificial Intelligence*, pages 674–679, 1981.

[6] J-Y. Bouguet. Pyramidal implementation of the lucas-kanade feature tracker description of the algorithm. Technical report, opencv documentation, Intel Corporation, Microprocessor Research Lab, 1999.

[7] G. Jaffré and A. Crouzil. Non-rigid object localization from color model using mean-shift. In *Proc. Int. Conf. Image Processing*, Sept. 2003.

[8] M.J. Swain and D.H. Ballard. Color indexing. *Int. J. Computer Vision*, 7, 1991.

[9] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, 2001.

[10] Y. Raja, S. J. McKenna, and S. Gong. Colour model selection and adaption in dynamic scenes. In *Proc. Europ. Conf. Computer Vision*, pages 460–474. Springer-Verlag, 1998.

[11] S. K. Zhou, Chellappa R., and Moghaddam B. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. Image Processing*, 13(11):1491–1506, 2004.

[12] Z. Zivkovic and B. Kröse. An EM-like algorithm for color-histogram-based object tracking. In *Proc. Conf. Comp. Vision Pattern Rec.*, 2004.

[13] K. Okuma, J. J. Little, and D. G. Lowe. Automatic rectification of long image sequences. In *Proc. Asian Conf. Computer Vision*, 2004.

[14] J-B. Hayet, J. Piater, and J. Verly. Robust incremental rectification of sport video sequences. In *Proc. British Machine Vision Conf.*, 2004.

[15] M. Black and P. Anandan. The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[16] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *J. Visual Com. and Image Representation*, 6(4):348–365, 1995.

[17] P.J. Huber. *Robust Statistics*. John Wiley and Sons, 1981.

[18] J-P. Tarel, S-S. Ieng, and P. Charbonnier. Using robust estimation algorithms for tracking explicit curves. In *Proc. Europ. Conf. Computer Vision*, pages 492–507, 2002.

[19] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New-York, 2001.

[20] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: multitarget detection and tracking. In *Proc. Europ. Conf. Computer Vision*, pages 28–39, 2004.

[21] E. Arnaud, E. Mémin, and B. Cernuschi-Frias. Conditional filters for image sequence based tracking - application to point tracking. *IEEE Trans. Image Processing*, 14(1):63–79, 2005.

[22] M. Isard and A. Blake. CONDENSATION–conditional density propagation for visual tracking. *Int. J. Computer Vision*, 29(1):5–28, 1998.

[23] Z. Khan, T. Balch, and F. Dellaert. An MCMC–based particle filter for tracking multiple interacting targets. In *Proc. Europ. Conf. Computer Vision*, pages 279–290, 2004.

[24] J. Vermaak, A. Doucet, and P. Pérez. Maintaining multi-modality through mixture tracking. In *Proc. Int. Conf. Computer Vision*, pages 1110–1116, October 2003.

[25] Y Cai, N. De Freitas, and J. J. Little. Robust visual tracking for multiple targets. In *Proc. Europ. Conf. Computer Vision*, 2006.

[26] J. Li, W. Ng, S. Godsill, and J. Vermaak. Online multitarget detection and tracking using sequential monte carlo methods. In *Proc. Int. Conf. Information Fusion*, pages 115–121, 2005.

[27] F. Pitié, S-A. Berrani, R. Dahyot, and A. Kokaram. Off-line multiple object tracking using candidate selection and the viterbi algorithm. In *Proc. Int. Conf. Image Processing*, 2005.

[28] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking – Linking identities using bayesian network inference. In *Proc. Conf. Comp. Vision Pattern Rec.*, pages II:2187–2194, 2006.

[29] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1):57–71, 2000.

[30] M. Isard and A. Blake. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. Europ. Conf. Computer Vision*, pages 893–908, 1998.

Centre de recherche INRIA Rennes – Bretagne Atlantique
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399