A Particle Filter for Joint Detection and Tracking of Color Objects

Jacek Czyz[a,*], Branko Ristic[b], Benoit Macq[a]

[a]Communications Laboratory, Université catholique de Louvain, Place du Levant 2, 1348 Louvain-la-Neuve, Belgium

[b]ISRD, DSTO, 200 Labs, PO Box 1500, Edinburgh, SA 5111, Australia

Abstract

Color is a powerful feature for tracking deformable objects in image sequences with complex backgrounds. The color particle filter has proven to be an efficient, simple and robust tracking algorithm. In this paper, we present a hybrid valued sequential state estimation algorithm, and its particle filter-based implementation, that extends the standard color particle filter in two ways. Firstly, target detection and deletion are embedded in the particle filter without relying on an external track initialization and cancellation algorithm. Secondly, the algorithm is able to track multiple objects sharing the same color description while keeping the attractive properties of the original color particle filter. The performance of the proposed filter are evaluated qualitatively on various real-world video sequences with appearing and disappearing targets.

Key words: visual tracking; particle filter; hybrid sequential estimation; multiple-target tracking

1 Introduction

Tracking moving objects in video sequences is a central concern in computer vision. Reliable visual tracking is indispensable is many emerging vision applications such as automatic video surveillance, human-computer interfaces and robotics. Traditionally, the tracking problem is formulated as sequential recursive estimation [1]: having an estimate of the probability distribution of the target in the previous frame, the problem is to estimate the target distribution in the new frame using all available prior knowledge and the new information brought by the new frame. The state-space formalism, where the current tracked object properties are described in an unknown state vector updated by noisy measurements, is very well adapted to model the tracking. Unfortunately the sequential estimation has an analytic solution under very restrictive hypotheses. The well known Kalman filter [2,3] is such a solution, and is optimal for the class of linear Gaussian estimation problems. The particle filter (PF), a numerical method that allows to find an approximate solution to the sequential estimation [4], has been successfully used in many target tracking problems [1] and visual tracking problems [5]. Its success, in comparison with the Kalman filter, can be explained by its capability to cope with multimodal measurements densities and non-linear observation models. In visual tracking, multi-modality of the measurement density is very frequent due to the presence of *distractors* – scene elements which has a similar appearance to the target [6]. The observation model, which relates the state vector to the measurements, is non-linear because image data (very redundant) undergoes feature extraction, a highly non-linear operation.

[*] Corresponding author. Tel. +32-10-479353; Fax: +32-10-472089.
Email address: czyz@tele.ucl.ac.be (Jacek Czyz).

1.1 Joint detection and tracking

Our work evolves from the adaptive color-based particle filter proposed independently by [7] and [8]. This color-based tracker uses color histograms as image features following the popular Mean-Shift tracker by Comaniciu *et al.* [9]. Since color histograms are robust to partial occlusion, are scale and rotation invariant, the resulting algorithm can efficiently and successfully handle non-rigid deformation of the target and rapidly changing dynamics in complex unknown background. However, it is designed for tracking a single object and uses an external mechanism to initialize the track. If multiple targets are present in the scene, but they are distinctive from each other, they can be tracked independently by running multiple instances of the color particle filter with different target models. In contrast, when several objects sharing the same color description are present in the scene (e.g. football game telecast, recordings of colonies of ants or bees, etc), the color particle filter approach fails either because particles are attracted by the different objects and the computed state estimates are meaningless, or particles tend to track only the best-fitting target – a phenomenon called *coalescence* [10]. In both cases alternative approaches must be found.

In this paper we develop a particle filter which extends the color particle filter of [7] and [8] by *integrating* the detection of new objects in the algorithm, and by tracking multiple similar objects. Detection of new targets entering the scene, i.e. track initialization, is embedded in the particle filter without relying on an external target detection algorithm. Also, the proposed algorithm can track multiple objects sharing the same color description and moving within the scene. The key feature in our approach is the augmentation of the state vector by a discrete-valued variable which represents the number of existing objects in the video sequence. This random variable is incorporated into the state vector and modeled as an M-state Markov chain. In this way, the problem of joint detection and tracking of multiple objects translates into a hybrid valued (continuous-discrete) sequential state estimation problem. We propose a conceptual solution to the joint detection and tracking problem in the Bayesian framework and we implement it using sequential Monte Carlo methods. Experimental results on real data show that the tracker, while keeping the attractive properties of the original color particle filter, can detect objects entering or leaving the scene; it keeps an internal list of observable objects (that can vary from 0 to a predefined number) without the need of external detection and deletion mechanisms.

1.2 Related work

Mixed or hybrid valued (continuous-discrete) sequential state estimation, and its particle-based solution, have been successful in many video sequence analysis problems. In [11], the proposed tracking algorithm switches between different motion models depending on a discrete label, included in the state vector, which encodes which one of a discrete set of motion models is active. Black and Jepson proposed a mixed state-space approach to gesture/expression recognition [12]. First several models of temporal trajectories are trained. Next the models are matched against new unknown trajectories using a PF-based algorithm in which the state vector contains a label of the model that matches the observed trajectory. The Bayesian Multiple-Blob tracker [13], BraMBLe, manages multiple blob tracking also by incorporating the number of visible objects into the state vector. The multi-blob observation likelihood is based on filter bank responses which may come from the background image or one of the object models. In contrast to BraMBLe, which requires background and foreground models, the method that we propose does not need a background estimation and can be used directly in camera moving sequences. Moreover, using color for describing the targets leads to small state vector size in comparison with contour tracking. The low dimensionality of the state-space permits the use of a smaller number of particles for the same estimation accuracy. This allows the algorithm to track many similar objects given some computational resources.

Classically the problem of visual tracking of multiple objects is tackled using data association techniques [14]. Rasmussen and Hager [6] use the Joint Probabilistic Data Association Filter (JPDAF) to track several *different* objects. Their main concern is to find correspondence between image measurements and tracked objects. Other approaches include [15],[16] where moving objects are modeled as *blobs*, i.e. groups of connected pixels, which are detected in the images using a combination of stereo vision and background modeling. MacCormick and Blake [10] studied particle-based tracking of multiple identical objects. They proposed an exclusion principle to avoid the coalescence onto the best fitting target when two targets come close to each other. This principle prevents a single image feature from being simultaneously reinforced by mutually exclusive hypotheses (either the image feature is generated by one target or by the other, but not both at the same time) . Our approach circumvents the need of the exclusion principle by integrating into the filter the target-merge operation (one target occludes the other) and target-split operation (the two targets are visible again) through the object existence variable.

Recently the mixture particle filter has been proposed [17]. In this work, the coalescence of multiple targets is avoided by maintaining the multi-modality of the state posterior density over time. This done by modeling the state density as a non-parametric mixture. Each particle receives, in addition to a weight, an indicator describing to which mixture component it belongs. A re-clustering procedure must be applied regularly to take into account appearing and disappearing modes.

1.3 Paper organization

The paper is organized as follows. In the next section we formulate joint detection and tracking of multiple targets as a sequential state estimation problem. First we explain how the discrete variable denoting the number of targets is modeled. Then the two step conceptual solution to the hybrid estimation problem is given. In Section 3 we present a numerical solution to the estimation problem which is obtained by a particle filter using color histograms as target features. Section 4 is devoted to experiments. Conclusions are given in the last section.

2 Sequential Recursive Estimation

The aim is to perform simultaneous detection and tracking of objects described by some image features (we chose the color histograms), in a video sequence $\mathbf{Z}_k = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k\}$, where $\mathbf{z}_j$, is the image frame at discrete-time (sequence) index $j = 1, \ldots, k$. This task is to be performed in a sequential manner, that is as the image frames become available over time. In the following, we first briefly review sequential recursive estimation for the case of a single target. The reader is referred to [18],[1] for more details. We then provide in detail the proposed formal recursive solution for multiple target tracking.

2.1 Bayesian estimation for single target tracking

In the state-space formalism, the state of the target is described by a state vector $\mathbf{x}_k$ containing parameters such as target position, velocity, angle, or size. The target evolves according to the following discrete-time stochastic model called the *dynamic model*

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \tag{1}$$

where $\mathbf{f}_{k-1}$ is a known function and $\mathbf{v}_{k-1}$ is the process noise. Equivalently, target evolution can be characterized by the *transition density* $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. The target state is related to the measurements via the *observation model*

$$\mathbf{z}_k = \mathbf{h}_{k-1}(\mathbf{x}_k, \mathbf{w}_k), \tag{2}$$

where $\mathbf{h}_{k-1}$ is a known function and $\mathbf{w}_{k-1}$ is the measurement noise. Again, the *observation density* $p(\mathbf{z}_k|\mathbf{x}_k)$ characterizes equivalently the relationship between the state vector and the measurement. Given the sequence of all available measurements $\mathbf{Z}_k = \{\mathbf{z}_i, i = 1, ..., k\}$, we seek $p(\mathbf{x}_k|\mathbf{Z}_k)$. The Bayesian estimation allows to find $p(\mathbf{x}_k|\mathbf{Z}_k)$ in a recursive manner, i.e. in terms of the posterior density at previous time step $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$. The conceptual solution is found in two steps. Using the transition density one can perform the prediction step:

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})d\mathbf{x}_{k-1}. \tag{3}$$

The prediction step makes use of the available knowledge of target evolution encoded in the dynamic model. The update step uses the measurement $\mathbf{z}_k$, available at time k, to update the predicted density:

$$p(\mathbf{x}_k|\mathbf{Z}_k) \propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1}). \tag{4}$$

These two steps, repeated for each frame, allow to compute recursively the state density for each frame.

2.2 Bayesian estimation for detection and multi-target tracking

In the case of multiple targets, each target i is characterized by one state vector $\mathbf{x}_{i,k}$ and one transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ if independent motion models are assumed. This results in a multi-object state vector $\mathbf{X}_k = (\mathbf{x}_{1,k}^T, ..., \mathbf{x}_{M,k}^T)^T$ where superscript T stands for the matrix transpose [19]. However the detection of appearing objects and deletion of disappearing objects is not integrated in the estimation process. To accommodate the Bayesian estimation with detection and varying number of targets, the state vector $\mathbf{X}_k$ is augmented by a discrete variable E_k, which we call *existence variable*, which denotes the number of visible or existing objects in the video frame k. The problem then becomes a jump Markov or hybrid estimation problem [20]. In the following we first describe how the existence variable E_k is modeled, we then present the proposed multi-target sequential estimation.

2.2.1 Object detection and deletion

The existence variable E_k is a discrete-valued random variable and $E \in \mathbb{E} = \{0, 1, \ldots, M\}$ with M being the maximum expected number of objects. The dynamics of this random variable is modeled by an M-state Markov chain, whose transitions are specified by an $(M+1)\times(M+1)$ transitional probability matrix (TPM) $\mathbf{\Pi} = [\pi_{ij}]$, where

$$\pi_{ij} = Pr\{E_k = j | E_{k-1} = i\}, \qquad (i, j \in \mathbb{E}) \tag{5}$$

is the probability of a transition from i objects existing at time $k-1$ to j objects at time k. The elements of the TPM satisfy $\sum_{j=1}^{M} \pi_{ij} = 1$ for each $i, j \in \mathbb{E}$. The dynamics of variable E is fully specified by the TPM and its initial probabilities at time $k = 0$, i.e. $\mu_i = Pr\{E_0 = i\}$, for $i = 0, 1, \ldots, M$.

For illustration, if we were to detect and track a single object (i.e. $M = 1$), the TPM is a 2×2 matrix given by:

$$\mathbf{\Pi} = \begin{bmatrix} (1-P_b) & P_b \\ P_d & (1-P_d) \end{bmatrix}$$

where P_b and P_d represent the probability of object "birth" (entering the scene) and "death" (leaving the scene), respectively. Similarly, for $M = 2$, a possible Markov chain which does not allow transitions from zero to two objects and from two to zero objects, is shown in Figure 1. The TPM of this model is given by:

$$\mathbf{\Pi} = \begin{bmatrix} (1-P_b) & P_b & 0 \\ P_d & (1-P_d-P_m) & P_m \\ 0 & P_r & (1-P_r) \end{bmatrix}$$

Again P_b, P_d, P_m and P_r are the design parameters. For higher values of M a similar model must be adopted.

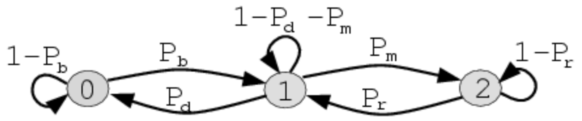


Fig. 1. A Markov chain of E_k variable for $M = 2$

2.2.2 Formal solution and state estimates

This section describes the conceptual solution to integrated detection and tracking of multiple objects in the sequential Bayesian estimation framework.

Let us first introduce a new state vector $\mathbf{y}_k$, which consists of variable E_k and the state vector $\mathbf{x}_{i,k}$ for each "existing" object i. The size of $\mathbf{y}_k$ depends on the value of E_k that is:

$$\mathbf{y}_k = \begin{cases} E_k & \text{if } E_k = 0 \\ [\mathbf{x}_{1,k}^T \ E_k]^T & \text{if } E_k = 1 \\ [\mathbf{x}_{1,k}^T \ \mathbf{x}_{2,k}^T \ E_k]^T & \text{if } E_k = 2 \\ \vdots & \vdots \\ [\mathbf{x}_{1,k}^T \ \ldots \ \mathbf{x}_{M,k}^T \ E_k]^T & \text{if } E_k = M \end{cases} \tag{6}$$

where $\mathbf{x}_{m,k}$ is the state vector of object $m = 1, \ldots, E_k$ at time k. Given the posterior density $p(\mathbf{y}_{k-1}|\mathbf{Z}_{k-1})$, and the latest available image $\mathbf{z}_k$ in the video sequence, the goal is to construct the posterior density at time k, that is $p(\mathbf{y}_k|\mathbf{Z}_k)$. This problem is an instance of sequential *hybrid* estimation, since one component of the state vector is discrete valued, while the rest is continuous valued.

Once the posterior pdf $p(\mathbf{y}_k|\mathbf{Z}_k)$ is known, the probability $P_m = Pr\{E_k = m|\mathbf{Z}_k\}$ that there are m objects in a video sequence at time k is computed as the marginal of $p(\mathbf{y}_k|\mathbf{Z}_k)$, i.e.:

$$P_m = \int \ldots \int p(\mathbf{x}_{1,k}, \ldots, \mathbf{x}_{m,k}, E_k = m|\mathbf{Z}_k) d\mathbf{x}_{1,k} \ldots d\mathbf{x}_{m,k} \tag{7}$$

for $m = 1, \ldots, M$. The case $m = 0$ is trivial, since in this case $p(\mathbf{y}_k|\mathbf{Z}_k)$ reduces to $Pr\{E_k = 0|\mathbf{Z}_k\}$. The MAP estimate of the number of objects at time k is then determined as:

$$\hat{m}_k = \arg \max_{m=0,1,\ldots,M} P_m. \tag{8}$$

This estimate provides the means for automatic detection of new object appearance and the existing object disappearance. The posterior pdfs of state components corresponding to individual objects in the scene are then computed as the marginals of pdf $p(\mathbf{x}_{1,k}, \ldots, \mathbf{x}_{\hat{m},k}, E_k = \hat{m}|\mathbf{Z}_k)$.

The two step procedure consisting of *prediction* and *update* is described in the following.

2.2.3 Prediction step

Suppose that m objects are present and visible in the scene with $0 \leq m \leq M$. In this case $E_k = m$ and the predicted state density can be expressed as:

$$p(\mathbf{x}_{1,k}, \ldots, \mathbf{x}_{m,k}, E_k = m|\mathbf{Z}_{k-1}) = \sum_{j=0}^{M} p_j \tag{9}$$

where, using notation

$$\mathbf{X}_k^j \equiv \mathbf{x}_{1,k} \ \ldots \ \mathbf{x}_{j,k}, \tag{10}$$

we have

$$p_j = \int p(\mathbf{X}_k^m, E_k = m|\mathbf{X}_{k-1}^j, E_{k-1} = j, \mathbf{Z}_{k-1}) \, p(\mathbf{X}_{k-1}^j, E_{k-1} = j|\mathbf{Z}_{k-1}) \, d\mathbf{X}_{k-1}^j \tag{11}$$

for $j = 0, \ldots, M$. Equation (9) is a prediction step because on its right hand side (RHS) features the posterior pdf at time $k-1$. Further simplification of (11) follows from

$$\begin{aligned} p(\mathbf{X}_k^m, &E_k = m | \mathbf{X}_{k-1}^j, E_{k-1} = j, \mathbf{Z}_{k-1}) = \\ &p(\mathbf{X}_k^m | \mathbf{X}_{k-1}^j, E_k = m, E_{k-1} = j) Pr\{E_k = m | E_{k-1} = j\}. \end{aligned} \tag{12}$$

Note that the second term on the RHS of (12) is an element of the TPM, i.e. $Pr\{E_k = m|E_{k-1} = j\} = \pi_{jm}$. Assuming that objects' states (kinematics, size parameters) are mutually independent, the first term of the RHS of (12) can be expressed as:

$$p(\mathbf{X}_k^m | \mathbf{X}_{k-1}^j, E_k = m, E_{k-1} = j) = \begin{cases} \prod_{i=1}^{m} p(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1}) & \text{if } m = j \\ \prod_{i=1}^{j} p(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1}) \prod_{i=j+1}^{m} p_b(\mathbf{x}_{i,k}) & \text{if } m > j \\ \prod_{i=1}^{j} \left[p(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1})\right]^{\delta_i} & \text{if } m < j \end{cases}$$

where

- $p(\mathbf{x}_{i,k}|\mathbf{x}_{i,k-1})$ is the transitional density of object i, defined by the object dynamic model, see (1). For simplicity, we assume independent motion models of the targets. In theory nothing prevents from creating joint motion models. However, this would require huge amounts of data to train the models.
- $p_b(\mathbf{x}_{i,k})$ is the initial object pdf on its appearance, which in the Bayesian framework is assumed to be known (subscript b stands for "birth"). For example, we can expect the object to appear in a certain region (e.g. along the edges of the image), with a certain velocity, length and width. If this initial knowledge is imprecise, we can model $p_b(\mathbf{x}_{i,k})$ with a uniform density.
- $\delta_1, \delta_2, \ldots, \delta_j$, which features in the case $m < j$, is a random binary sequence, such that $\delta_i \in \{0, 1\}$ and $\sum_{i=1}^{j} \delta_i = m$. Note that the distribution of the δ_i

(which may depend on $\mathbf{x}_i$) reflect our knowledge on disappearance of object i. Again, $\Pr\{\delta_i = 0|\mathbf{x}_i\}$ might be higher if the object $\mathbf{x}_i$ is close to the edges of the image. If this knowledge is imprecise, we can model these distributions by uniform distributions.

2.2.4 Update step

The update step results from the application of the Bayes rule and formally states:

$$p(\mathbf{X}_k^m, E_k = m|\mathbf{Z}_k) = \frac{p(\mathbf{z}_k|\mathbf{X}_k^m, E_k = m)\, p(\mathbf{X}_k^m, E_k = m|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})}, \tag{13}$$

where $p(\mathbf{X}_k^m, E_k = m|\mathbf{Z}_{k-1})$ is the prediction density given by (9) and $p(\mathbf{z}_k|\mathbf{X}_k^m, E_k = m)$ is the image likelihood function. From image $\mathbf{z}_k$ we extract a set of features $q_{i,k}$, for each of $i = 1, ..., m$ objects, and use them as measurements (the choice of the features is discussed in Sec.3.2). Assuming these features are mutually independent from one object to another, we can replace $p(\mathbf{z}_k|\mathbf{X}_k^m, E_k = m)$ with

$$p(q_{1,k}, \ldots, q_{m,k}|\mathbf{X}_k^m, E_k = m) = \prod_{i=1}^{m} p(q_{i,k}|\mathbf{x}_{i,k}). \tag{14}$$

As the image region where $q_{i,k}$ is computed is encoded in the state vector $\mathbf{x}_{i,k}$ there is no problem of associating measurements to state vectors. The assumption that object features are independent for different objects holds only if the image region inside which we compute object features are non-overlapping. In the overlapping case, the same image feature is attributed to two different objects, which is not realistic. In order to prevent several objects from being associated to the same feature, we impose that the distance between two objects cannot be smaller than a threshold. Therefore when a target A passes in front of a target B and occludes it, only one target will be existing. The drawback is that when B reappears, there must be some logic that says “OK this is B again”. For the time being, B will be simply re-detected and considered as a new object. To solve this problem one option is to have a “track management system” on top of the presented algorithm. This system would store the position, heading and speed (and possibly other useful attributes) of the object when it disappears behind the occlusion and compare it to the position of a new object that appears in the vicinity (both in space and time). Using simple heuristics the correspondence between the disappearance/apparition could be established in many cases. Another option is to use a multi-camera setup. The proposed PF would output a list of 2D target positions for each camera view that are collected to a central data association module. This module would perform data association and output 3D positions of the targets. The problem of occlusion is largely avoided in this case since when a target is occluded in one view, it is often visible in the other.

The described conceptual solution for simultaneous detection and tracking of a varying number of objects next has to be implemented. Note that the hybrid sequential estimation is non-linear even if the dynamic and observation models are linear [1], a Kalman filter is therefore inappropriate for solving the problem and one must look for approximations.

3 Color-based Particle Filter

The general multi-target sequential estimation presented in the previous section can be adapted to different applications by an appropriate choice of the dynamic and observation models. In visual tracking the state vector characterizes the target (region or shape parameters). The observation model reflects which image features will be used to update the current state estimate. As in [9],[7],[8], we use color histograms extracted from the target region as image feature. This choice is motivated by ease of implementation, efficiency and robustness. However, the general framework for joint detection and tracking can be adapted to other observation features such as appearance models [21].

3.1 Dynamic model

The state vector of a single object typically consists of kinematic and region parameters. We adopt the following state vector

$$\mathbf{x}_k = [x_k \; y_k \; H_k \; W_k]^T, \tag{15}$$

where (x_k, y_k) denotes the center of the image region (in our case a rectangle) within which the computation of object's color histogram is carried out; H_k and W_k denote the image region parameters (in our case its width and height); superscript T in (15) stands for the matrix transpose. Object motion and the dynamics of its size are modeled by a random walk, that is the state equation is linear and given by:

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{w}_{k-1}. \tag{16}$$

Process noise $\mathbf{w}_{k-1}$ in (16) is assumed to be white, zero-mean Gaussian, with the covariance matrix $\mathbf{Q}$. Other motion models (e.g. constant velocity) and higher dimensional state vectors (e.g. one could include the aspect ratio change rate of the image region rectangle in the state vector) might be more appropriate depending on the application.

3.2 Color measurement model

Following [9],[7],[8], we do not use the entire image $\mathbf{z}_k$ as the measurement, but rather we extract from the image the color histogram q_k, computed inside the rectangular region whose location and size are specified by the state vector $\mathbf{x}_k$: the center of the region is in (x_k, y_k); the size of the region is determined by (H_k, W_k).

We adopt the Gaussian density for the likelihood function of the measured color histogram as follows:

$$p(q_k|\mathbf{x}_k) \propto \mathcal{N}(D_k; 0, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{D_k^2}{2\sigma^2}\} \tag{17}$$

where $D_k = \text{dist}[q^*, q_k]$ is the distance between (i) the reference histogram q^* of objects to be tracked and (ii) the histogram q_k computed from image $\mathbf{z}_k$ in the region defined by the state vector $\mathbf{x}_k$. The standard deviation σ of the Gaussian density in (17) is a design parameter.

Suppose $q^* = \{q^*(u)\}_{u=1,\dots,U}$ and $q_k = \{q_k(u)\}_{u=1,\dots,U}$ are the two histograms calculated over U bins. We adopt the distance D_k between two histograms derived in [9] from the Bhattacharyya similarity coefficient, defined as:

$$D_k = \sqrt{1 - \sum_{u=1}^{U} \sqrt{q^*(u)\ q_k(u)}}. \tag{18}$$

The computation of histograms is typically done in the RGB space or HSV space [8]. A weighting function, which assigns smaller weights to the pixels that are further away from the region center, is often applied in computing the histograms. In this way the reliability of the color distribution is increased when boundary pixels belong to the background or get occluded.

In the framework of multiple object, we must extract multiple histograms from the measurement $\mathbf{z}_k$ as specified by Equation (14). Therefore, based on (17), the multi-object observation likelihood becomes

$$p(\mathbf{z}_k|\mathbf{X}_k^m, E_k = m) \propto \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{1}{2\sigma^2}\sum_{i=1}^{m} D_{i,k}^2\} \tag{19}$$

where $D_{i,k} = \text{dist}[q^*, q_{i,k}]$ is the distance between i-th object color histogram and the reference color histogram. Note that $q_{i,k}$ is the color histogram computed from $\mathbf{z}_k$ in the region specified by $\mathbf{x}_{i,k}$.

We point out that the described measurement likelihood function is not necessarily restricted to color video sequences - recently it has been applied for detection and tracking of objects in monochromatic FLIR imagery [22].

3.3 Particle Filter

Particle filters are sequential Monte Carlo techniques specifically designed for sequential Bayesian estimation when systems are non-linear and random elements are non-Gaussian. The hybrid estimation presented in the previous section, with the dynamic model and the highly non-linear observation model described above, is carried out using a particle filter. Particle filters approximate the posterior density $p(\mathbf{y}_k|\mathbf{Z}_k)$ by a weighted set of random samples or particles. In our case, a particle of index n is characterized by a certain value of E_k^n variable and the corresponding number of state vectors $\mathbf{x}_{i,k}^n$ where $i = 1, \dots, E_k^n$, i.e.

$$\mathbf{y}_k^n = \left[E_k^n, \mathbf{x}_{1,k}^n, \dots, \mathbf{x}_{E_k^n,k}^n\right] \qquad (n = 1, \dots, N)$$

where N is the number of particles. The pseudo-code of the main steps of this filter (single cycle) are presented in Table 1. The input to the PF are the particles at time $k-1$ and the image at time k; the output are the particles at time k.

Next we describe in more detail each step of the algorithm of Table 1.

- The first step in the algorithm represents random transition of E_{k-1}^n to E_k^n based on the TPM $\mathbf{\Pi}$. This is done by implementing the rule that if $E_{k-1}^n = i$ then E_k^n should be set to j with probability π_{ij}. See for example [1] (Table 3.9) for more details.
- Step 2.a of Table 1 follows from equation (13). If $E_{k-1}^n = E_k^n$, then we draw $\mathbf{x}_{i,k}^n \sim p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{x}_{i,k-1}^n)$ for $i = 1, \ldots, E_k^n$. In our implementation we use the transitional prior for this purpose, that is $\mathbf{x}_{i,k}^n \sim p(\mathbf{x}_k|\mathbf{x}_{i,k-1}^n)$. If the number of objects is increased from $k-1$ to k, i.e. $E_{k-1}^n < E_k^n$, then for the objects that continue to exist we draw $\mathbf{x}_{i,k}^n$ using the transitional prior (as above), but for the newborn objects we draw particles from $p_b(\mathbf{x}_k)$. Finally if $E_{k-1}^n > E_k^n$, we select at random E_k^n objects from the possible E_{k-1}^n, with equal probability. The selected objects continue to exist (the others do not) and for them we draw particles using the transitional prior (as above).
- Step 2.b follows from equations (14) and (19). In order to perform its role of a detector, the particle filter computes its importance weights based on the likelihood ratio

$$L_k(m) = \prod_{i=1}^{m} \frac{p(q_{i,k}|\mathbf{x}_{i,k})}{p(q_{i,k}^B|\mathbf{x}_{i,k})} \tag{20}$$

where $q_{i,k}^B$ is the color histogram of the image background computed in the region specified by $\mathbf{x}_{i,k}$. Using (19), the likelihood ratio can be computed for each existing particle n as

$$L_k^n(E_k^n) = \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{E_k^n}\left[\left(D_{i,k}^n\right)^2 - \left(D_{i,k}^{n,B}\right)^2\right]\right\} \tag{21}$$

where

$$D_{i,k}^n = \text{dist}\left[q^*, q_{i,k}^n(\mathbf{z}_k)\right] \tag{22}$$

is the distance between the reference histogram q^* and the histogram $q_{i,k}^n$ computed from $\mathbf{z}_k$ in the region specified by $\mathbf{x}_{i,k}^n$ and

$$D_{i,k}^{n,B} = \text{dist}\left[q^*, q_{i,k}^n(\mathbf{z}_B)\right] \tag{23}$$

is the distance between the reference histogram q^* and the histogram $q_{i,k}^n(\mathbf{z}_B)$ computed at the same position but from the background image $\mathbf{z}_B$. The unnormalized importance weights are computed for each particle as:

$$\tilde{w}_k^n = \begin{cases} 1, & \text{if } E_k^n = 0 \\ L_k^n(E_k^n), & \text{if } E_k^n > 0. \end{cases} \tag{24}$$

Note that if the distance sum $\sum_{i=1}^{E_k^n} D_{i,k}^n$ in (21) is smaller than the background distance sum $\sum_{i=1}^{E_k^n} D_{i,k}^{n,B}$, then the weight $\tilde{w}_{i,k}^n$ is greater than 1, and this particle has a better chance of survival in the resampling step.

Strictly speaking, the computation of L_k requires that the color histogram of the image background is known at every location of the image. In many cases this is impractical (especially when the camera is moving and the background is varying), hence we approximate the distance $D_{i,k}^B$ between the target histogram and the background histogram as constant over all the image and for $i = 1, ..., M$. Introducing the constant $C_B = \exp\{(D_{i,k}^B)^2/2\sigma^2\}$

the likelihood ratio becomes

$$L_k = \prod_i^m \exp\{\frac{1}{2\sigma^2}(D^B_{i,k})^2\} \exp\{-\frac{1}{2\sigma^2}\sum_{i=1}^m D^2_{i,k}\} \tag{25}$$

$$= (C_B)^m \exp\{-\frac{1}{2\sigma^2}\sum_{i=1}^m D^2_{i,k}\}. \tag{26}$$

We thus treat the constant C_B as a design parameter as in [7]. C_B is adapted to take into account the similarity between the target and the background histogram.

An additional condition must be added. To prevent a region histogram from being attributed to two overlapping objects and to avoid the appearance of multiple objects on the same image region, the weight of the particle is set to zero if two objects are too close to each other, i.e. $\tilde{w}^n_k = 0$ if $(x^n_{j,k} - x^n_{i,k})^2 + (y^n_{j,k} - y^n_{i,k})^2 < R^2$ where R^2 is a constant fixed by the user.

- For the resampling step 5, standard $O(N)$ algorithms exist, see for example Table 3.2 in [1].
- The output of the PF (step 6) is carried out for the reporting purposes, and consists of estimation of the number of objects $\hat{m}$ and the estimation of objects' states. The number of objects is estimated based on (8), where $Pr\{E_k = m|\mathbf{Z}_k\}$ is computed in the PF as:

$$Pr\{E_k = m|\mathbf{Z}_k\} = \frac{1}{N}\sum_{n=1}^N \delta(E^n_k, m) \tag{27}$$

and $\delta(i,j) = 1$, if $i = j$, and zero otherwise (Kroneker delta). The estimate of the state vector of object $i = 1, \ldots, \hat{m}$ is then

$$\hat{\mathbf{x}}_{i,k|k} = \frac{\sum_{n=1}^N \mathbf{x}^n_{i,k}\ \delta(E^n_k, i)}{\sum_{n=1}^N \delta(E^n_k, i)}. \tag{28}$$

Table 1

Particle filter pseudo-code (single cycle)

```
$[\{\mathbf{y}_k^n\}_{n=1}^N] = \mathrm{PF}[\{\mathbf{y}_{k-1}^n\}_{n=1}^N, \mathbf{z}_k]$
(1) Transitions of $E_{k-1}$ variable (random transition of the number of existing objects):
    $[\{E_k^n\}_{n=1}^N] = \mathrm{ETrans}\ [\{E_{k-1}^n\}_{n=1}^N, \mathbf{\Pi}]$
(2) FOR $n = 1 : N$
      a. Based on $(E_{k-1}^n, E_k^n)$ pair, draw at random $\mathbf{x}_{1,k}^n, \ldots, \mathbf{x}_{E_k^n,k}^n$;
      b. Evaluate importance weight $\tilde{w}_k^n$ (up to a normalizing constant) using (24).
(3) END FOR
(4) Normalize importance weights
      a. Calculate total weight: $t = \mathrm{SUM}\ [\{\tilde{w}_k^n\}_{n=1}^N]$
      b. FOR $n = 1 : N$
           • Normalize: $w_k^n = t^{-1}\tilde{w}_k^n$
         END FOR
(5) Resample:
    $[\{\mathbf{y}_k^n, -, -\}_{n=1}^N] = \mathrm{RESAMPLE}\ [\{\mathbf{y}_k^n, w_k^n\}_{n=1}^N]$
(6) Compute the output of the PF
```

4 Experimental results

Experiments were conducted on several real world image sequences. The sequences can be found at *http://euterpe.tele.ucl.ac.be/Tracking/pf.html*. The practical details of the implemented algorithm are described next.

The transitional probability matrix is simplified as described in Section 2.2.1:

only transitions from m_{k-1} objects at time $k-1$ to $m_{k-1} \pm 1$ objects at k are allowed, with probability 0.05. In this way the TPM is a tri-diagonal matrix, with approximately 5% of the particles in the state with $E_k = m_{k-1} \pm 1$. The probability that the number of objects remains unchanged is accordingly set to 0.90. This simplification of the TPM means that if two objects appear at the same time, the estimate of the object number $\hat{m}$ will be incremented in two steps.

The distribution of newborn particles $p_b(\mathbf{x}_{i,k})$ was adopted to be a uniform density over the state vector variables (i.e. no prior knowledge as to where the objects will appear). Finally, the color histograms were computed in the RGB color space using 8x8x8 bins as in [8]. The target histogram is acquired using a few training frames. In each frame, the target region is selected manually and a histogram is computed from the region. The target histogram is obtained by averaging the histograms obtained for each frame.

The number of particles required by the filter depends on the selected value of M and the prior knowledge on where the objects are likely to appear (this knowledge is modeled by $p_b(\mathbf{x}_{i,k})$). For $M = 1$, the filter with up to $N = 150$ particles achieves adequate accuracy both for detection and estimation. For $M = 6$ identical objects, it was necessary to use $N = 5000$ particles in order to detect rapidly new appearing objects. This number can certainly be decreased using a better prior $p_b(\mathbf{x}_{i,k})$ or a more adequate dynamic model depending on the application. Also, that the transition density that we use as proposal density is not optimal [23]. As we are primarily interested in testing the viability of the algorithm, we used the simplest models and the least prior knowledge in order to stay independent of a particular application.

In the first example the objective is to detect and track two different objects (i.e. two humans with different color histograms) in a video sequence recorded with a surveillance camera. The image resolution is 435×343 pixels. The first person (person A) wears a white shirt, with a black tie and his pants are black. The second person (person B) is in a blue t-shirt. There are 200 image frames available for detection and tracking, with a camera moving slowly in order to follow person A. The estimated probabilities of existence of both person A and B are shown in Figure 3. Eight selected image frames are displayed in Figure 2. Here we effectively run two particle filters in parallel, each tuned (by the reference histogram) to detect and track its respective object. Each filter is using 150 particles, with $\sigma = 0.8$ and $C_B = 30$. Person A appears in the first image frame and continues to exist throughout the video sequence. The particle filter detects it in the frame number 14: the probability of existence of person 1 jumps to the value of 1 between frame 14 and 16, as indicated in Figure 3. A detected object/person is indicated in each image by a white rectangle, located at the estimated object position. Person B enters the scene (from the left) in frame 50 and is detected by the PF in frame 60. Frame 79 is noteworthy: here person B partially occludes person A, and this is very well reflected in the drop of the probability of existence for person A; see again Figure 3. In frame 160, person B leaves the scene, and hence its probability of existence drops to zero; person A is continued to be tracked until the last frame.

In the second example the aim is to detect and track three identical red and white colored cans in a cluttered background. The complete sequence is 600 frames long and can be viewed on the web site given above. Figure 4 shows an interesting event: one can is passing in front of the second. The filter is using

$N = 1000$ particles with parameters $\sigma = 0.6$ and $C_B = 70$, the image size is 640 x 480. The two cans appear at frame 101 and are detected at frame 151. At frame 183, one can is occluded by the other. The second object is deleted by the filter at frame 187. At frame 226, the second can is again visible and the filter detects its presence at frame 232. Note that the filter does not perform data association. As the object are not distinguishable, it cannot maintain the "identity" of the cans when they merge and subsequently split.

In the third example the objective is to detect (as they enter or leave the scene) and track the soccer players of the team in red and black shirts (with white-colored numbers on their back). Figure 5 displays 12 selected frames of this video sequence, with a moving camera. The image resolution is 780×320 pixels. The filter is using $N = 5000$ particles, with parameters $\sigma = 0.6$ and $C_B = 80$. We observe that initially five red players are present in the scene. Frames 4, 9, 35 and 67 show that the first, second, third and fourth player are being detected respectively. Hence $\hat{m}_{67} = 4$. At frame 99, the first detected player leaves the scene. It is deleted by the filter at frame 102, $\hat{m}_{102}$ switches back to 3. At the same time, another player is entering the scene. It is detected at frame 141. One of the players leaves the scene at the top of frame 173, and subsequently it is deleted. This demonstrates a quick response of the particle filter to the change of the number of objects. All three remaining players are tracked successfully until the last frame.

The algorithm processing time is of course related to the number of particles needed to make the algorithm work. When the number of objects is small (i. e. 1,2 objects), with "gentle" motion (i.e. the dynamical model is accurately describing the motion), then the number of particles is below 500. In that case, our C++ implementation can run at 15 frames/second on a 2.8 GHz

CPU . However, in the football sequence showed in the experiments, there are 5 objects to detect and track and the required number of particles is then 5,000. In that case the algorithm works at about 1frame/sec. Thus the main drawback of the proposed approach is that the number of particles increases with the number of objects (i.e. the size of the state vector). An excellent analysis of the relationship between the state vector size and the number of particles was presented in [24] and can be summarized as follows: using a smart proposal density in the PF, this relationship can be made linear, otherwise it tends to be exponential.

5 Conclusion

The paper presented a formal recursive estimation method for joint detection and tracking of multiple objects having the same feature description. This formal solution was then implemented by a particle filter using color histograms as object features. The performance of the detecting and tracking algorithm was then tested on several real world sequences. From the results, the algorithm can succesfully detect and track many identical targets. It can handle non-rigid deformation of targets, partial occlusions and cluttered background. Also the experimental results confirm that the method can be successfully applied even when the camera is moving. The key hypothesis in the adopted approach is that the background is of a sufficiently different color structure than the objects to be tracked. However, to alleviate this problem different observation features can be used in addition to color as for example appearance models [21] or contours.

This basic algorithm can be improved in several ways. For example, color histograms can be computed in different regions of the target (face, shirt, pants, etc) in order to take into account the topological information [7]. Also, the number of required particles could be reduced by adopting a better proposal density for existing particles and a better prior density of appearing objects.

References

[1] B. Ristic, S. Arulampalam, N. Gordon, Beyond the Kalman filter: Particle filters for tracking applications, Artech House, 2004.

[2] R. E. Kalman, A new approach to linear filtering and prediction problems, Transaction of the ASME Journal of Basic Engineering 82 (1960) 35–45.

[3] G. Welch, G. Bishop, A introduction to the Kalman fitler, Technical report TR 95-041.

[4] A. Doucet, J. F. G. de Freitas, N. J. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer, New York, 2001.

[5] M. Isard, A. Blake, Visual tracking by stochastic propagation of conditional density, in: Proc. European Conf. Computer Vision, 1996, pp. 343–356.

[6] C. Rasmussen, G. Hager, Probabilistic data association methods for tracking complex visual objects, IEEE Trans. on Pattern Analysis and Machine Intelligence 23 (6) (2001) 560–576.

[7] P. Pérez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, in: A. H. et al. (Ed.), Proc. European Conf. Computer Vision (ECCV), Springer-Verlag, 2002, pp. 661–675, lNCS 2350.

[8] K. Nummiaro, E. Koller-Meier, L. Van-Gool, An adaptive color-based particle filter, Image and Vision Computing 21 (2003) 99–110.

[9] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: Proc. IEEE Conf. Comp. Vision Pattern Recog., Hilton Head, SC, 2000, pp. II:142–149.

[10] J. MacCormick, A. Blake, A probabilistic exclusion principle for tracking multiple objects, International Journal on Computer Vision 39 (1) (2000) 57–71.

[11] M. Isard, A. Blake, A mixed-state condensation tracker with automatic model-switching, in: Proc. Int. Conf. Computer Vision, 1998, pp. 107–112.

[12] M. J. Black, A. D. Jepson, Recognizing temporal trajectories using the condensation algorithm, in: Proc. of the 3rd Int. Conf. Automatic Face and Gesture Recognition, 1998, pp. 16–21.

[13] M. Isard, J. MacCormick, BraMBLe: a bayesian multiple blob tracker, in: Proc. Int. Conf. Computer Vision, 2001, pp. 34–41.

[14] I. Cox, A review of statistical data association techniques for motion correspondence, Int. J. of Computer Vision 10 (1) (1993) 53–66.

[15] I. Haritaoglu, D. Harwood, L. Davis, W4s: A real-time system for detecting and tracking people in 2 1/2-d, in: European Conf. on Computer vision, 1998, pp. 877–892.

[16] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder: Real-time tracking of the human body, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 780–785.

[17] J. Vermaak, A. Doucet, P. Perez, Maintaining multi-modality through mixture tracking, in: Int. Conf. on Computer Vision, 2003, pp. 1110–1116.

[18] S. Arulampalam, S. Maskell, N. J. Gordon, T. Clapp, A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking, IEEE Transactions of Signal Processing 50 (2) (2002) 174–188.

[19] C. Hue, J.-P. L. Cadre, P. Pérez, Tracking multiple objects with particle filtering, IEEE Trans. on Aerospace and Electronic Systems 38 (32) (2002) 791–812.

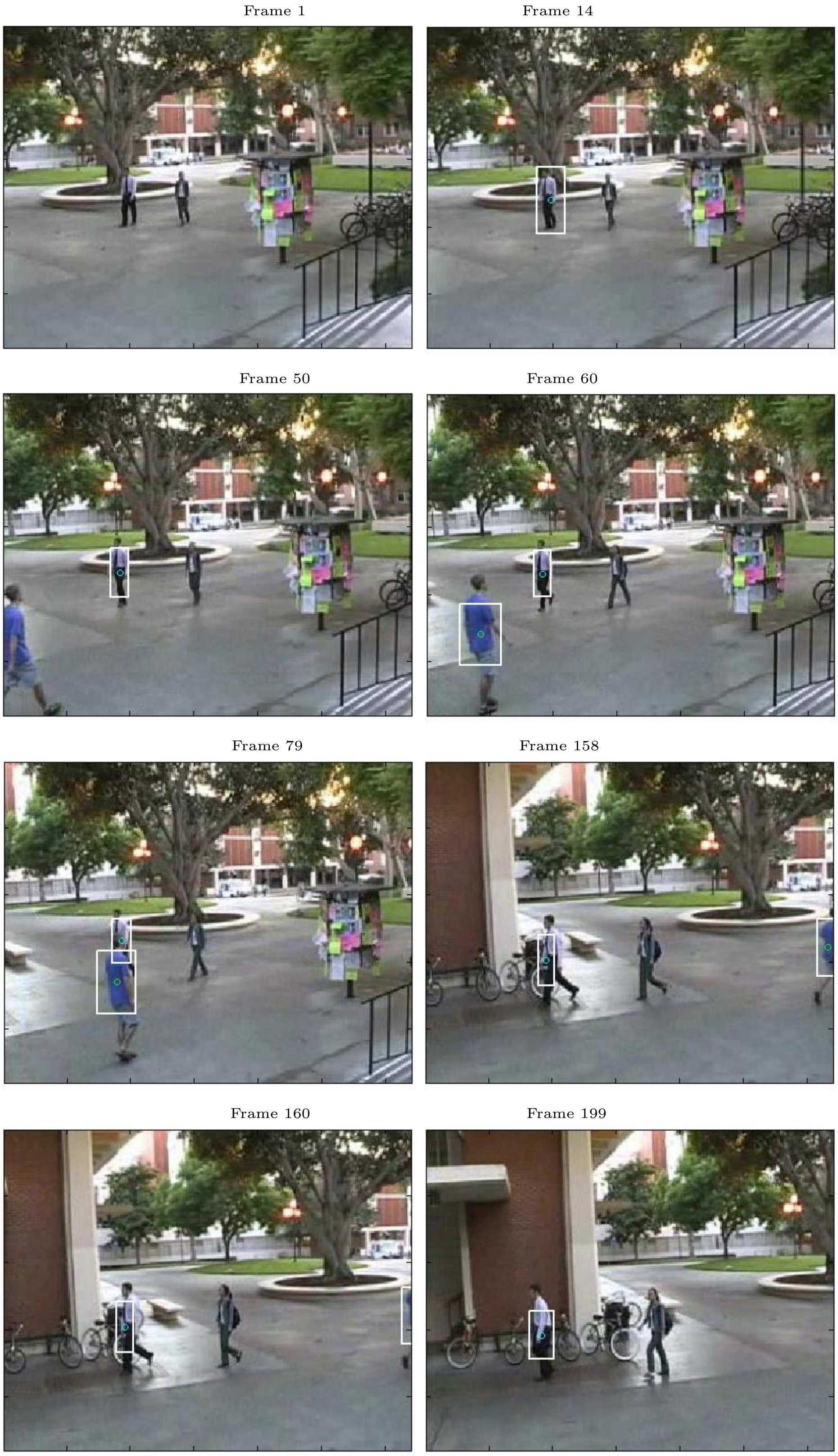
[20] G. D. Forney, The Viterbi algorithm, Proc. of the IEEE 61 (1973) 268–278.

[21] S. Zhou, R. Chellappa, B. Moghaddam, Visual tracking and recognition using appearance-adaptive models in particle filters, IEEE Transactions on Image Processing 13 (11) (2004) 1434–1456.

[22] A. Yilmaz, K. Shafique, M. Shah, Target tracking in airborne forward looking infrared imagery, Image and Vision Computing 21 (2003) 623–635.

[23] A. Doucet, S. Godsill, C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, Statistics and Computing 10 (3) (2000) 197–208.

[24] F. Daum, J. Huang, Curse of dimensionality and particle filters, in: Proc. IEEE Aerospace Conf., Big Sky, Montana, USA, 2003.



Frame 1

Frame 14

Frame 50

Frame 60

Frame 79

Frame 158

Frame 160

Frame 199

Fig. 2. *Surveillance camera sequence: detected and tracked persons are marked with a rectangle*

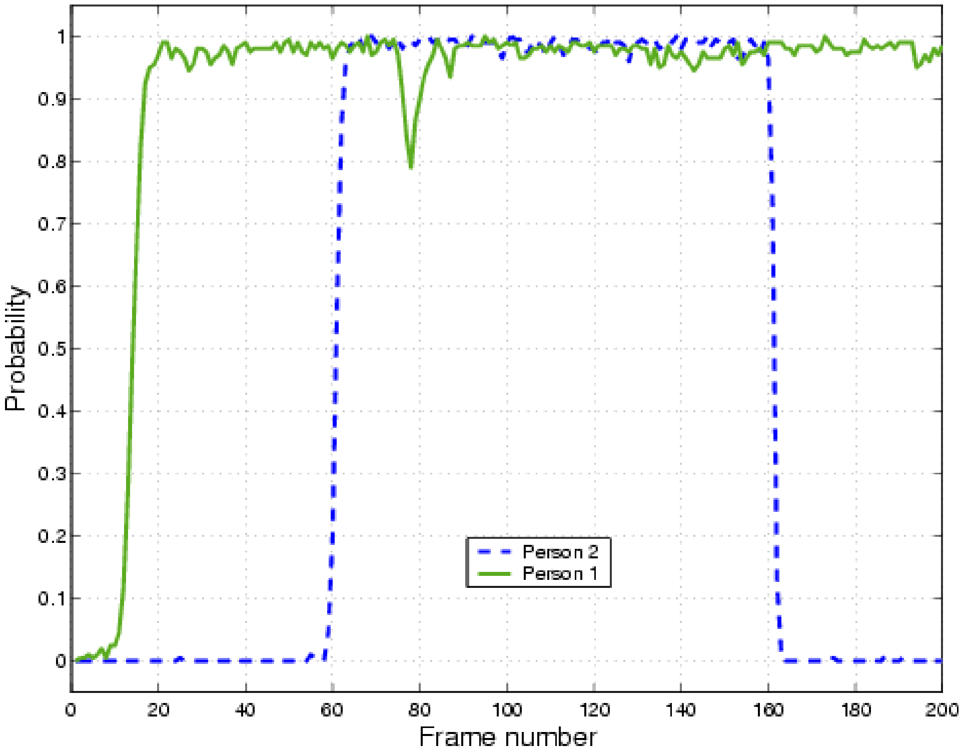
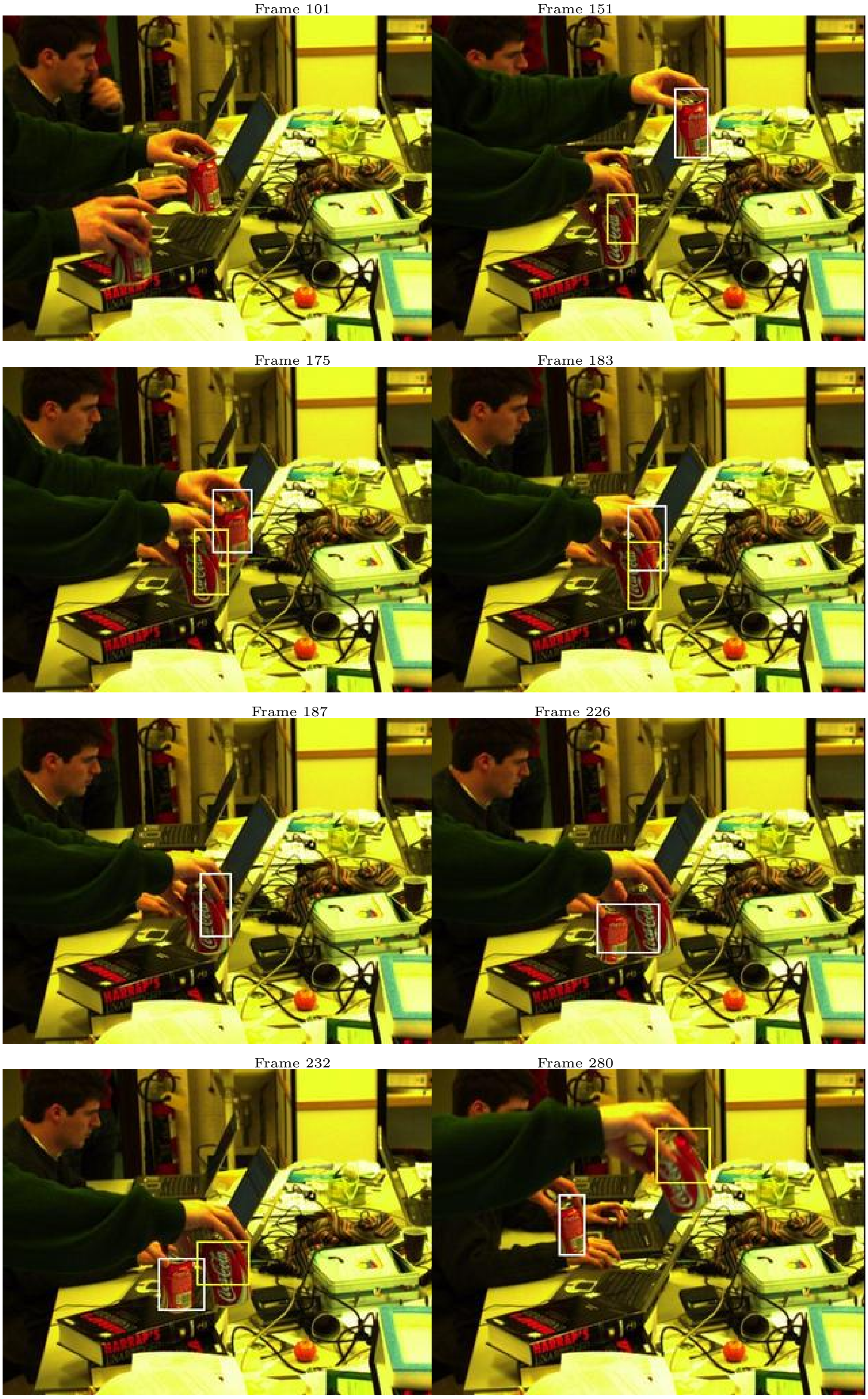


Fig. 3. *The probability of existence for object 1 and 2 in the video sequence of Figure 2*



Frame 101

Frame 151

Frame 175

Frame 183

Frame 187

Frame 226

Frame 232

Frame 280

Fig. 4. *Can sequence: detected and tracked cans are marked with a rectangle*

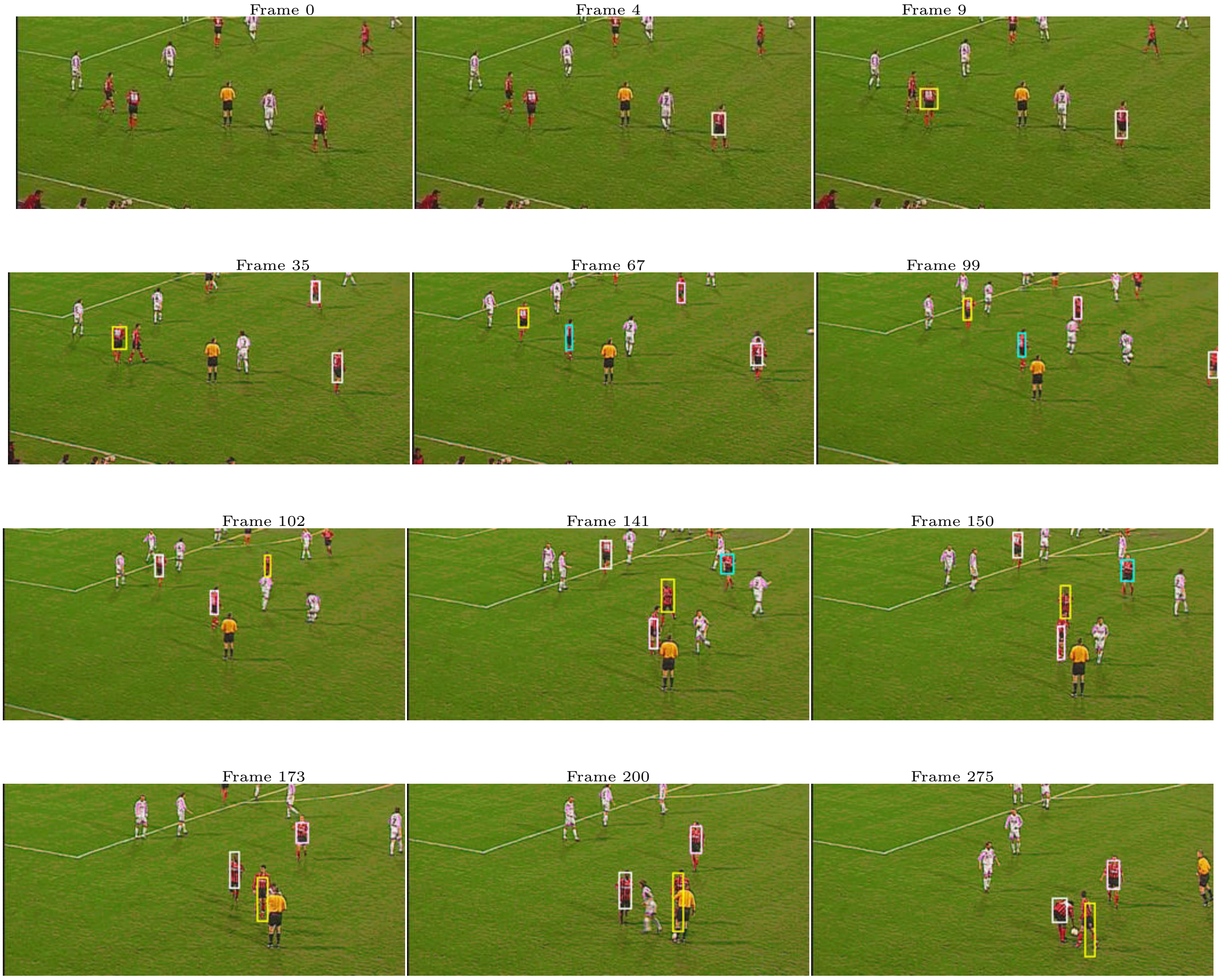


Fig. 5. *Football sequence: detected and tracked players are marked with a rectangle*