Introducing Fuzzy Spatial Constraints in a Ranked Partitioned Sampling for Multi-object Tracking

Nicolas Widynski[1,2], Séverine Dubuisson[2], and Isabelle Bloch[1]

[1] Télécom ParisTech, CNRS UMR 5141 LTCI, Paris, France
[2] UPMC - LIP6, Paris, France

Abstract. Dealing with multi-object tracking in a particle filter raises several issues. A first essential point is to model possible interactions between objects. In this article, we represent these interactions using a fuzzy formalism, which allows us to easily model spatial constraints between objects, in a general and formal way. The second issue addressed in this work concerns the practical application of a multi-object tracking with a particle filter. To avoid a decrease of performances, a partitioned sampling method can be employed. However, to achieve good tracking performances, the estimation process requires to know the ordering sequence in which the objects are treated. This problem is solved by introducing, as a second contribution, a ranked partitioned sampling, which aims at estimating both the ordering sequence and the joint state of the objects. Finally, we show the benefit of our two contributions in comparison to classical approaches through two multi-object tracking experiments and the tracking of an articulated object.

1 Introduction

Sequential Monte Carlo methods, also known as particle filters, have been widely used in the vision community. Their natural dispositions for tracking purposes, their reliability to deal with non linear systems and their easiness of implementation have certainly contributed to this success. Dealing with possibly numerous interacting objects requires to represent the interactions between objects and to employ an efficient algorithm whatever the number of objects to track. We focus here on these two classical issues.

As a first contribution, we propose to introduce fuzzy spatial constraints between objects into the particle filter. This allows easily modeling potentially complex interactions between objects. To our knowledge, spatial constraints in a particle filter framework have only been used in specific ways, in a non fuzzy formalism (e.g. [1]). Fuzzy spatial constraints have shown a real interest in various domains, such as clustering [2], brain segmentation in 3D MRI images [3] or graph reasoning over fuzzy attributes [4].

The adaptation of particle filters to track several objects has been extensively addressed in the literature, in many different ways. Among these, the authors

G. Bebis et al. (Eds.): ISVC 2010, Part I, LNCS 6453, pp. 393–404, 2010.

in [5] proposed a Jump Markov System to model the number of objects, the association hypothesis between observations and objects and the indivual states. In [6], the authors use one particle filter and model interactions between objects and measures using a Joint Probabilistic Data Association Filter (JPDAF) framework. In [7], the distribution of the association hypotheses is computed using a Gibbs sampler. On another hand, as pointed out in [8], the importance sampling, used in particular in particle filters, suffers from the problem of the curse of dimensionality. This means that the particle filter requires a number of particles that exponentially increases with the number of objects. This renders the practical use of a particle filter for multiple object tracking difficult as soon as the number of objects is greater than three. Therefore, the authors in [9,10] proposed a particle filter that avoids this additional cost using a partitioned sampling strategy, based on a principle of exclusion (i.e. specifying that a measurement may be associated to at most one object). This is performed by partitioning the state space, typically considering one element of the partition per object, according to a specific ordering of the objects that we call scenario, and to select particles, using a weighted sampling, that are the most likely to fit with the real state of the object. The considered order matters since it can lead to unsuitable behaviors of the filter, such as loosing tracks, for example when the first considered object is hidden. In fact, as pointed out in [11] and as we will detail further in this article, using a specific order to estimate the state may also have many bad effects, which can be explained by a phenomenon of impoverishment of the particle set. In [11], the joint filtering distribution is represented by a mixture model, in which each mixture component describes a specific order and is estimated using a partitioned sampling technique. This idea has also been used in [12] for multi-cue fusion purposes. However, the number of particles allocated to each component is fixed, that may degrade performances of the filter when the chosen orders are not relevant [11]. In this article, we propose to jointly estimate the order in which objects are processed and states of the objects, in a so called ranked partitioned sampling strategy. This allows us to consider the whole set of possible orders and to automatically prune irrelevant scenarios.

This paper is organized as follows. Section 2 presents the fuzzy spatial constraint formalism and its introduction into a probabilistic framework. Section 3 describes the multi-object tracking based on a particle filtering modeling. Then in Section 4, we recall the principle of the partitioned sampling before introducing our ranked partitioned sampling in Section 5. We finally show results in Section 6 by considering two classical multi-object tracking experiments and the tracking of an articulated object.

2 Modeling Fuzzy Spatial Constraints

In this section, we propose to model explicitly interactions between objects via fuzzy spatial relations defined over one, two or more objects indicating to which degree the relation is satisfied. They will be considered as constraints the objects should satisfy during the tracking process, to a non-zero degree, and are therefore called fuzzy spatial constraints. Each relation is considered as a linguistic variable, taking a small number of linguistic values [13]. The granularity of this representation can be defined by the application. The semantics of each linguistic value is defined by a fuzzy set on the variable domain. Fuzzy spatial constraints may be defined by unary fuzzy operators, such as the concept of the size of an object (which may take the values *small*, *medium*, *large*, ...); by binary operators, such as the concept of relative orientation (*is to the right of*, *is to the left of*, ...); by ternary operators, such as the concept of local disposition (*is the first of*, *is in the middle of*, *is the last of*); and more generally n-any operators. In this paper, we focus on binary, ternary and quaternary operators, considering concepts of intersection, distance, angle and alignment. Fuzzy spatial constraints may be fixed during the tracking process (the topology of the configuration of the objects is fixed, although known imprecisely), may evolve during the time (fuzzy spatial concepts gradually change their values), or may be defined over time (considering imprecise fuzzy spatio-temporal constraints). Here we only consider time-independent spatial relations but we believe that the two last types of constraints may be of interest too.

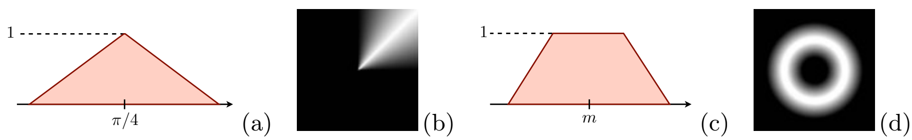


Fig. 1. Spatial fuzzy relations illustrated (a,c) in the variable domain and (b,d) in the image domain with respect to the center point. Figures (a,b) represent the value *north-east* of the concept of orientation and figures (c,d) the value *medium distance m* of the concept of distance.

Let $\mathbf{x}_t^*$ be an hypothetic state of an object and $\tilde{\mathbf{x}}_t = (\tilde{\mathbf{x}}_t^1, \ldots, \tilde{\mathbf{x}}_t^L)$ be the vector state of L objects already processed at time t. We now define a fuzzy membership function $\mu(\mathbf{x}_t^*; \tilde{\mathbf{x}}_t) \in [0, 1]$ which describes to which degree an object configuration $\mathbf{x}_t^*$ satisfies the spatial constraints imposed by $\tilde{\mathbf{x}}_t$. Denoting by K the number of spatial constraints we consider, we define μ as:

$$\mu(\mathbf{x}_t^*; \tilde{\mathbf{x}}_t) = \mathop{\Xi}_{k=1}^{K} \mu^k(\mathbf{x}_t^*; \tilde{\mathbf{x}}_t) \tag{1}$$

with Ξ a fusion operator, for example a t-norm (fuzzy conjunction) [14], and $\mu^k \in [0, 1]$ the membership function of the k^{th} spatial constraint. For example, considering a binary fuzzy relation, μ^k is defined as:

$$\mu^k(\mathbf{x}_t^*; \tilde{\mathbf{x}}_t) = \mathop{\psi}_{l=1}^{L} \mu_l^k(\mathbf{x}_t^*; \tilde{\mathbf{x}}_t^l) \tag{2}$$

with ψ a fusion operator, for example a t-norm, and $\mu_l^k \in [0, 1]$ the membership function of the k^{th} spatial constraint between the current object and object l.

The shape of the function μ^k may be fixed by the application. In our experiments, we consider a triangular one to generate the orientation *north-east* and a trapezoidal one for the distance *medium*, respectively illustrated in Figures 1(a) and 1(b). In a similar way, considering ternary constraints, μ^k is defined as:

$$\mu^k(\mathbf{x}_t^*; \tilde{\mathbf{x}}_t) = \mathop{\psi}_{l_1=1}^{L} \mathop{\psi}_{l_2=l_1+1}^{L} \mu_{l_1,l_2}^k(\mathbf{x}_t^*; \tilde{\mathbf{x}}_t^{l_1}, \tilde{\mathbf{x}}_t^{l_2}) \tag{3}$$

with $\mu_{l_1,l_2}^k \in [0,1]$ the membership function of the k^{th} spatial constraint between the current object and objects l_1 and l_2. Finally, we introduce the fuzzy spatial constraints into a probabilistic framework, by defining the function $\phi(\mathbf{x}_t^*; \tilde{\mathbf{x}}_t)$ as:

$$\phi(\mathbf{x}_t^*; \tilde{\mathbf{x}}_t) \propto \mu(\mathbf{x}_t^*; \tilde{\mathbf{x}}_t)^\gamma \tag{4}$$

with $\gamma \in \mathbb{R}^+$ a fixed parameter and $\mu(\mathbf{x}_t^*; \tilde{\mathbf{x}}_t)$ the membership function of the spatial constraints defined in Equation 1. Examples of constraints will be defined in Section 6. The function ϕ allows passing from a possibilistic semantic to a probabilistic one and will be integrated in a dynamic model in a particle filtering framework (see Section 4).

3 Particle Filtering for Multi-object Tracking

In this article, we consider that the number M of objects is known. Let us consider a classical filtering problem and denote by $\mathbf{x}_t \in \mathbb{X}$ the hidden joint state of a stochastic process at time t: $\mathbf{x}_t = (\mathbf{x}_t^1, \ldots, \mathbf{x}_t^M)$ with $\mathbf{x}_t^i \in \mathbb{X}^*$ the unknown state of the i^{th} object, and by $\mathbf{y}_t \in \mathbb{Y}$ the measurement state extracted from the image sequence. The temporal evolution of $\mathbf{x}_t$ and the measurement equation are given by $\mathbf{x}_t = f_t(\mathbf{x}_{t-1}, \mathbf{v}_t)$ and $\mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{w}_t)$, where $\mathbf{v}_t$ and $\mathbf{w}_t$ are independent white noises. The non-linear Bayesian tracking consists in estimating the *posterior* filtering distribution $\Pr(\mathbf{x}_t|\mathbf{y}_{1:t})$ through a non-linear transition function f_t and a non-linear measurement function h_t. The resulting filtering density function can be expressed by $p(\mathbf{x}_t|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mathbf{x}_t) \int_{\mathbb{X}} p(\mathbf{x}_t|\mathbf{x}_{t-1})\, p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1}$. Particle filters are used to approximate the *posterior* density function (*pdf*) by a weighted sum of N Dirac masses $\delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$ centered on hypothetic state realizations $\{\mathbf{x}_t^{(n)}\}_{n=1}^N$ of the state $\mathbf{x}_t$, also called particles. Then, the filtering distribution $\Pr(d\mathbf{x}_t|\mathbf{y}_{1:t})$ is recursively approximated by the empiric distribution $P_N(d\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$, where $\{\mathbf{x}_t^{(n)}\}$ is the n^{th} particle and $w_t^{(n)}$ its weight. If an approximation of $\Pr(d\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ is known, the process is divided into three main steps:

1. The diffusion step consists in estimating $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ by propagating the particle swarm $\{(\mathbf{x}_{t-1}^{(n)}), w_{t-1}^{(n)}\}_{n=1}^N$ using an importance function $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_t)$.
2. The update step then computes new particle weights using the new observation $\mathbf{y}_t$, as: $w_t^{(n)} \propto w_{t-1}^{(n)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(n)}) p(\mathbf{x}_t^{(n)}|\mathbf{x}_{t-1}^{(n)})}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_t)}$, such that $\sum_{i=1}^N w_t^{(n)} = 1$.

3. Resampling techniques are employed to avoid particle degeneracy problems.

When needed, the data association problem may be handled by estimating both the objects and a vector of visibility of those [9,15]. This involves a measurement process dependent on the visibility of the objects. This point will be discussed in Sections 5 and 6.1.

The presented particle filter algorithm uses, by essence, an importance sampling procedure while simulating according to a density function $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_t)$. In practice, this means in particular that an increase in the dimension of $\mathbf{x}_t$ systematically induces an increase in the variance of the particle weights, leading to possibly fatal impoverishment of the particle set. In the case of multi-object tracking, where $\mathbf{x}_t^i$ and $\mathbf{x}_t^j$ are two states evolving in the same space $\mathbb{X}^*$, it has been shown [10] that N^2 particles are necessary to achieve the same level of tracking performance than when tracking a single object with N particles. To tackle the dimensionality problem, the authors in [10] propose, instead of directly sampling from the joint configuration of the objects, to decompose the vector state of the objects by partitioning the state space, and then handling one object at a time. This is called partitioned sampling and presented in the next section.

4 Partitioned Sampling (PS)

The Partitioned Sampling (PS), introduced in [9,10], divides the joint state space $\mathbb{X}$ into a partition of M elements, i.e. one by object, and for each of them, applies the transition (dynamics) and performs a weighted resampling operation.

Weighted Resampling. A weighted resampling operation transforms a particle set $\{\mathbf{x}_t^{(n)}, w_t^{(n)}\}_{n=1}^N$ into another one while keeping the distribution intact. Weights $\{\rho^{(n)}\}_{n=1}^N$ are called importance weights and are designed such that $\rho^{(n)} = g(\mathbf{x}_t^{(n)}) / \sum_{m=1}^N g(\mathbf{x}_t^{(m)})$. The strictly positive function g is called *weighting function*, and aims at resampling the particles set according to the peaks of g. Finally the particle set $\{\tilde{\mathbf{x}}_t^{(n)}, w_t^{(n)}/\rho^{(n)}\}_{n=1}^N$ is obtained by selecting particles $\{\mathbf{x}_t^{(m)}\}_{m=1}^N$ with probabilities $\{\rho^{(m)}\}_{m=1}^N$.

Partitioned Resampling. Denoting by $\sim g_i$ the weighted resampling operation of the i^{th} object, by $\sim$ the resampling procedure according to the particle weights $\{w_t^{(n)}\}_{n=1}^N$, and by f_i the dynamics process of the object i, possibly conditioned by objects already generated $\mathbf{x}_t^{1:i-1}$, the partitioned sampling operation is summarized in Figure 2(a).

Although any weighting function g_i should asymptotically keep the *posterior* unchanged, the objective of this step is to obtain an accurate representation of the *posterior*. Then, considering a factorization of the likelihood such that it allows us to deal with each object independently, i.e. $p(\mathbf{y}_t|\mathbf{x}_t) = \prod_{i=1}^M p_i(\mathbf{y}_t|\mathbf{x}_t^i)$, then the likelihood $h_i = p_i(\mathbf{y}_t|\mathbf{x}_t^i)$ of the object i appears to be a natural choice and leads to the diagram proposed in Figure 2(b).

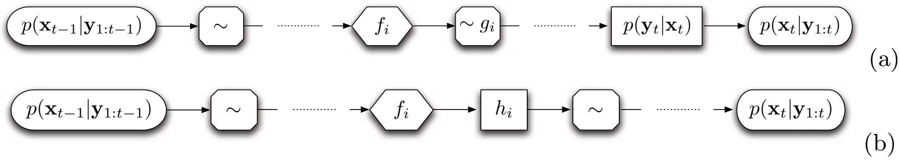


Fig. 2. (a) Diagram of the Partitioned Sampling procedure and (b) diagram of the Partitioned Sampling procedure using the likelihood as weighting function

Additionally, we propose to integrate the fuzzy spatial constraints modeled in Section 2. The simplest way to do it consists in introducing the interaction density function defined in Equation 4 into the dynamical model, leading to $f_i = p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)\,\phi(\mathbf{x}_t^i; \mathbf{x}_t^{1:i-1})$. This model can be viewed as the pairwise Markov Random Field *prior* used in [16,1,12]. However, in a more general perspective, it is often impossible to directly generate samples from $\phi(\mathbf{x}_t^i; \mathbf{x}_t^{1:i-1})$. Then, we consider $f_i = p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$ whereas the likelihood integrates the interaction term, i.e., $h_i = p_i(\mathbf{y}_t|\mathbf{x}_t^i)\phi(\mathbf{x}_t^i; \mathbf{x}_t^{1:i-1})$. This procedure does not affect the *posterior* since it can be seen as an importance sampling step [1].

Discussion. The partitioned sampling is a very efficient sampling method since, by alleviating the dimension problem, it considerably reduces the computation cost. However, as discussed in [11], the order of the considered objects has a direct impact on the performance of the tracker. This is due to the M successive weighting resampling procedures performed by the algorithm. Hence, objects placed at early stages will be prone to more impoverishment effects than the others. On the other side, objects placed at the end may suffer from a lack of diversity even before being considered, which may also lead to tracking errors.

The method presents an additional difficulty. If occlusions occur, one may quite rightly handle visible objects first, and hence adopt a dynamic order strategy of the objects. The solution proposed in [9] is called *Branched Partitioned Sampling* (BPS), and consists in adding to the global estimation a vector of visibility, and then recursively grouping together particles with an identic realization of this vector, generating an hypothesis tree. This solution has however a major drawback: considering a tracking problem with M objects, the method possibly divides particles into $M!$ hypotheses, which looses the interest of the partitioned sampling since the particles no longer try to survive over a large set of N particles but over possible irrelevant sets of $N/M!$ elements.

The *Dynamic Partitioned Sampling* (DPS), proposed in [11], uses a mixture model to represent the *posterior* distribution. Each mixture component represents a specific order of processing of the objects. In their experiments, the authors used M permutation sets, deterministically defined, each one owning N/M particles. This strategy improves Partitioned Sampling results since it alleviates impoverishment effects, especially when occlusions occur. However, using a fixed small subset of possible permutations might not be robust. Moreover, splitting particles into several sets has the same drawback than the Branched Partitioned Sampling, since particles evolve into subsets.

To overcome these problems, we propose a new sampling strategy, called *Ranked Partitioned Sampling* (RPS), which jointly estimates the state of the objects and the estimation of the order of processing of the objects. Objects with highest confidence are considered in earlier stages of the scenarios. Each scenario is then confronted to each of the others, implicitly pruning unlikely branches. The adapative choice of the order of processing aims at limiting the impoverishment effect.

5 Ranked Partitioned Sampling (RPS)

Let $\mathbf{o}_t = (\mathbf{o}_t^1, \ldots, \mathbf{o}_t^M)$ be an ordered sequence of processing, i.e. a permutation over M objects. Then $\mathbf{o}_t^i$ is a random variable that indicates the position of the object i in this ordered sequence. We call a scenario the inverse permutation $\mathbf{o}_t^{-1}$ considered at a particular instant t, which we denote by $\mathbf{s}_t = (\mathbf{s}_t^1, \ldots, \mathbf{s}_t^M)$. Hence the k^{th} component of a scenario is defined such that $\mathbf{s}_t^k \triangleq \sum_{i=1}^{M} i\, \delta_{\mathbf{o}_t^i}^k$, with δ_a^b the Kronecker function that equals 1 if $a = b$, 0 otherwise, and indicates the object processed at the step k. We first consider fixed probabilities of transition of the positions:

$$p(\mathbf{s}_t^h = i | \mathbf{s}_{t-1}^k = i) \triangleq p(\mathbf{o}_t^i = h | \mathbf{o}_{t-1}^i = k) \triangleq \alpha_{k,h} \quad \forall i \in \{1, \ldots, M\} \tag{5}$$

By first considering objects placed in the earliest stages at time $t-1$, the joint transition distribution of $\mathbf{o}_t$ is:

$$p(\mathbf{o}_t | \mathbf{o}_{t-1}) = p\left(\mathbf{o}_t^{s^1} | \mathbf{o}_{t-1}^{s^1}\right) \prod_{k=2}^{M} p\left(\mathbf{o}_t^{s^k} | \mathbf{o}_{t-1}^{s^k} \triangleq k, \mathbf{o}_t^{s^1}, \ldots, \mathbf{o}_t^{s^{k-1}}\right) \tag{6}$$

with $\mathbf{s}^k \triangleq \mathbf{s}_{t-1}^k$, the time subscript being omitted to simplify notations. The last conditional distribution in Equation 6 depends on the probabilities of transitions of positions defined in Equation 5 and inaccessible positions of already considered objects:

$$p\Big(\mathbf{o}_t^{s^k} = h | \mathbf{o}_{t-1}^{s^k}, \{\mathbf{o}_t^{s^u}\}_{u=1}^{k-1}\Big) = \left[1 - \sum_{j=1}^{k-1} \delta_{\mathbf{o}_t^{s^j}}^h\right] \left[\alpha_{k,h} + \frac{1}{M-k+1} \sum_{j=1}^{k-1} \alpha_{k,\mathbf{o}_t^{s^j}}\right] \tag{7}$$

The first term in the product ensures that the probability is set to 0 if the position h has already been assigned, whereas the last term uses probabilities of transition of the assigned positions to balance the distribution in a uniform way. We decompose the joint transition density so that $p(\mathbf{x}_t, \mathbf{o}_t | \mathbf{x}_{t-1}, \mathbf{o}_{t-1}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{o}_t) p(\mathbf{o}_t | \mathbf{o}_{t-1})$. Conditioned by the sequence order defined by $\mathbf{o}_t$, the transition density of the vector state $\mathbf{x}_t$ is decomposed considering first the objects placed in the earliest stages:

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{o}_t) \triangleq \prod_{k=1}^{M} p(\mathbf{x}_t^{s_t^k} | \mathbf{x}_{t-1}^{s_t^k}, \mathbf{x}_t^{s_t^1}, \ldots, \mathbf{x}_t^{s_t^{k-1}}) \tag{8}$$

However, following the choices made in Section 4, we set the dynamic process to $f_{\mathbf{s}_t^k} = p(\mathbf{x}_t^{\mathbf{s}_t^k} | \mathbf{x}_{t-1}^{\mathbf{s}_t^k})$. In the same way, the likelihood is defined as $p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{o}_t) \triangleq \prod_{k=1}^{M} p_{\mathbf{s}_t^k}(\mathbf{y}_t|\mathbf{x}_t^{s_t^k})$. To summarize, at a time t, for each particle, the algorithm first generates a scenario, and then, at position k of the process (scenario), it proposes a new state of the object $\mathbf{s}_t^k$ using dynamics, and computes the likelihood before resampling all the particles. The approximation of the joint filtering distribution of $\mathbf{x}_t, \mathbf{o}_t$ is obtained once the M positions have been computed. By setting $h_{\mathbf{s}_t^k} = p_{\mathbf{s}_t^k}(\mathbf{y}_t|\mathbf{x}_t^{\mathbf{s}_t^k})\phi(\mathbf{x}_t^{\mathbf{s}_t^k}; \mathbf{x}_t^{{\mathbf{s}_t}^{1}:{\mathbf{s}_t}^{k-1}})$, we obtain the diagram in Figure 3.

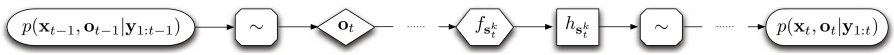


Fig. 3. Diagram of the Ranked Partitioned Sampling procedure using the likelihood as weighting function

There are many ways to deal with hidden objects. A current choice is to estimate a visibility vector and then proposing scenarios according to it. However, in this paper, we implicitly consider that an object at a position i is always less visible than a object at position $i - k$, with $0 < k < i$.

6 Experiments

6.1 People Tracking

We consider a public sequence [17], from which we extracted 260 frames where three pedestrians walk and occult each other. Let $\mathbf{x}_t^i = (\mathrm{x}_t^i, \mathrm{y}_t^i)^T$ be the unknown state of object i, with $(\mathrm{x}_t^i, \mathrm{y}_t^i)^T$ the $2D$ center of a person. Dimension of the rectangles surrounding persons are fixed. The dynamics is a random walk, i.e. $\mathbf{x}_t^i = f_t(\mathbf{x}_{t-1}^i, \mathbf{v}_t^i) = \mathbf{x}_{t-1}^i + \mathbf{v}_t^i$ with $\mathbf{v}_t^i \sim \mathcal{N}(\mathbf{0}_{2\times 1}, \Sigma)$. For the BPS, we modeled a visibility vector, where the transition probability from state *visible* to state *hidden* was fixed to 0.2, and the transition probability from state *hidden* to state *visible* to 0.5. These probabilities have been fixed empirically, in a way that they give priority to the state *visible*, although being flexible enough to deal with sudden occlusions. For the RPS, we implicitly consider the visibility of an object by its position in the processing order, then no spatial constraint is necessary for this first experiment. For both methods, visible objects are considered first, although in the RPS the visibility vector is not modeled since the scenario vector determines the visibility of an object. The likelihood of an object is based on a distance between a color model histogram and a candidate histogram defined by the particle [18]. However, only the visible part of the object is considered to avoid penalizing hidden or partially hidden objects.

Results using $N = 500$ particles are illustrated in Figure 4. Rectangles in red correspond to the estimation of objects (i.e. the Monte Carlo expected value).

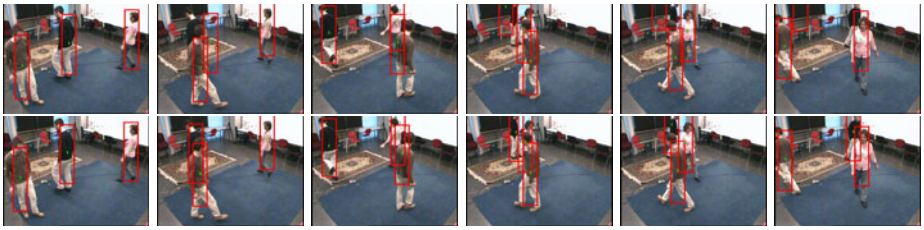
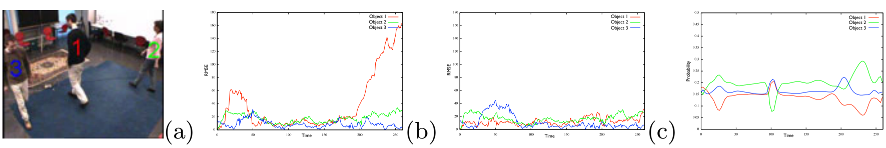


Fig. 4. People tracking results at times 5, 24, 99, 205, 212 and 259. First row: branched PS [9], second row: ranked PS (proposed approach).

As mentionned in Section 4, in the BPS, the particles may be divided into $M!$ sets, which may maintain scenarios where the visibility hypotheses are wrong. Moreover, the visibility vector is not well adapted in the case where the number of objects is greater than two, since it does not solve anymore the data association problem. These two points explain the difference of the results obtained by the BPS and the RPS (see e.g. second and last images). Figures 5(b) and 5(c) show the overall superiority of the RPS over the BPS. Figure 5(d) presents the *posterior* probabilities obtained by the RPS for a person to be considered first in the processing order induced by $\mathbf{o}_t$, where a low probability indicates that the person is likely to be partially hidden. We can appreciate the probabilities estimated for example at times 24, 99, 205 and 212 that are consistent with the sequence (Figure 4).



(a) (b) (c) (d)

Fig. 5. (a) Indices of the pedestrians present in the scene (b) RMSE of the branched PS, (c) RMSE of the ranked PS (proposed approach) and (d) *posterior* probabilities obtained by the RPS for a person to be considered first in the processing order

6.2 Ant Tracking

This test sequence has been successfully studied in [1], using a MCMC-Based particle filtering approach. The state of the object i, $\mathbf{x}_t^i = (\mathrm{x}_t^i, \mathrm{y}_t^i, \theta_t^i)^T$, contains the $2D$ position $(\mathrm{x}_t^i, \mathrm{y}_t^i)^T$ and the object orientation θ_t^i. Dynamics of position and orientation are random walks. We used for this experiment an exclusion fuzzy spatial constraint: using a fuzzy semi-trapeze, two ants must not overlap more than 10% of their own areas, and from 5% the degree of satisfaction of the constraint starts to decrease. The likelihood is a simple background substraction.

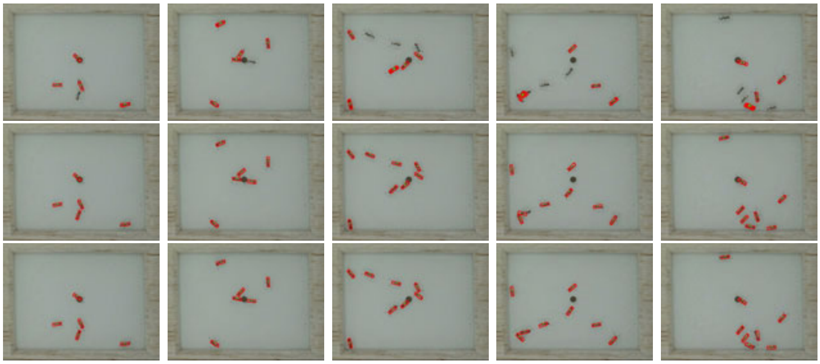


Fig. 6. Ant tracking results. First row: RPS without spatial constraints, second row: PS with spatial constraints and last row: RPS with spatial constraints.

Figure 6 shows results using a RPS with and without spatial constraints, and a PS with spatial constraints. Estimated positions of the ants are represented in red. $N = 500$ particles were used for this sequence of 750 frames. The benefit of using a simple spatial constraint is very clear here (several ants are not tracked without spatial constraints, while they are successfully tracked with such constraints). The PS and the RPS give global comparable results since all possible processing orders lead to almost identical results. Then the RPS performs as well as the PS when the order does not significantly matter and the number of objects is small enough to not suffer from an impoverishment effect.

6.3 Hand Tracking

We finally consider a problem of tracking an articulated object. The state of the object i, $\mathbf{x}_t^i = (\mathrm{x}_t^i, \mathrm{y}_t^i, \theta_t^i)^T$, contains its $2D$ center position $(\mathrm{x}_t^i, \mathrm{y}_t^i)^T$ and orientation θ_t^i. Each finger shape is fixed and expressed by vectors of 6 $2D$-control points, located on the basis of fingers, on the middle and on the fingertips. Dynamics of position and orientation are random walks. The difficulty of this application is that fingers may be partially or totally hidden. We then would like to track the hand by preserving a global consistency of the shape using fuzzy spatial constraints. Although they might be automatically learnt, we consider here fixed spatial relations between the fingers. For instance, we used four fixed fuzzy spatial constraints: angle, distance, alignment and exclusion. We defined two values of the binary constraint of angle, with an uncertainty expressed by a trapezoidal template of length support $\pi/4$: *nearly* $-\pi/8$, and *nearly* $\pi/8$; two values of the binary constraint of distance: *close* and *far*; one ternary or quaternary (it depends on the number of objets already processed) constraint of alignment: using a linear regression, fingers cannot move away from a fixed distance threshold; and one binary constraint of exclusion: no overlap between fingers is allowed. The likelihood is obtained by computing gradient values over normal lines of regular points of a finger B-spline [19].

Fig. 7. Hand tracking results. First row: RPS without spatial constraints, second row: PS with spatial constraints and last row: RPS with spatial constraints.

The results are illustrated in Figure 7 and were obtained using $N = 2000$ particles, over a sequence of 800 frames. Results using a simple partitioned sampling strategy were generated using a random sequence order (in this particular sequence, fixed sequence order gave not quite as good results). RPS without spatial constraints fails as soon as a finger is hidden. The RPS obtains better results than the PS thanks to the estimation of the scenario, which allows estimating first fingers that are trusted, the visible ones, and then the other ones, which are then more constrained by the fuzzy spatial relations.

7 Conclusion

This article presents two contributions. First, we introduced fuzzy spatial constraints into a multi-object tracking based on particle filtering. This novel information allows us to easily handle constraints between objects in a unified framework. As a second contribution, the multi-object particle filter uses a ranked partitioned sampling strategy, which, like the partitioned sampling, tackles the problem of dimensionality by sequentially performing a weigthed resampling step in single object state spaces. Moreover, the simulation order proposed in the RPS is adaptive, which makes the tracking more robust and alleviates the impoverishment effect, while keeping a computation time identical to the PS one.

References

1. Khan, Z., Balch, T., Dellaert, F.: MCMC-Based particle filtering for tracking a variable number of interacting targets. IEEE Transactions Pattern Analysis and Machine Intelligence 27, 1805–1918 (2005)
2. Pham, D.: Fuzzy clustering with spatial constraints. In: IEEE International Conference on Image Processing, vol. II, pp. 65–68 (2002)
3. Colliot, O., Camara, O., Bloch, I.: Integration of fuzzy spatial relations in deformable models - Application to brain MRI segmentation. Pattern Recognition 39, 1401–1414 (2006)

4. Fouquier, G., Atif, J., Bloch, I.: Local reasoning in fuzzy attribute graphs for optimizing sequential segmentation. In: Escolano, F., Vento, M. (eds.) GbRPR 2007. LNCS, vol. 4538, pp. 138–147. Springer, Heidelberg (2007)
5. Doucet, A., Vo, B., Andrieu, C., Davy, M.: Particle filtering for multi-target tracking and sensor management. In: 5th International Conference on Information Fusion, pp. 474–481 (2002)
6. Schulz, D., Burgard, W., Fox, D., Cremers, A.: Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In: IEEE International Conference on Robotics & Automation, pp. 1665–1670 (2001)
7. Hue, C., Le Cadre, J.P., Prez, P.: Sequential Monte Carlo methods for multiple target tracking and data fusion. IEEE Transactions on Signal Processing 50, 309–325 (2002)
8. MacKay, D.J.C.: Introduction to Monte Carlo methods. In: Jordan, M.I. (ed.) Learning in Graphical Models. NATO Science Series, pp. 175–204. Kluwer Academic Press, Dordrecht (1998)
9. MacCormick, J., Blake, A.: A probabilistic exclusion principle for tracking multiple objects. International Journal of Computer Vision, 39, 57–71 (2000)
10. MacCormick, J., Isard, M.: Partitioned sampling, articulated objects, and interface-quality hand tracking. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 3–19. Springer, Heidelberg (2000)
11. Smith, K., Gatica-Perez, D.: Order matters: A distributed sampling method for multi-object tracking. In: British Maschine Vision Conference, pp. 25–32 (2004)
12. Duffner, S., Odobez, J., Ricci, E.: Dynamic partitioned sampling for tracking with discriminative features. In: British Maschine Vision Conference (2009)
13. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning. Information Sciences 8, 199–249 (1975)
14. Dubois, D., Prade, H.: Fuzzy Sets and Systems: Theory and Applications. Academic Press, Inc., Orlando (1980)
15. Pérez, P., Vermaak, J.: Bayesian tracking with auxiliary discrete processes. application to detection and tracking of objects with occlusions. In: ICCV 2005 Workshop on Dynamical Vision, Beijing, China, pp. 190–202 (2005)
16. MacCormick, J., Blake, A.: Spatial dependence in the observation of visual contours. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, pp. 765–781. Springer, Heidelberg (1998)
17. Fleuret, F., Berclaz, J., Lengagne, R., Fua, P.: Multi-camera people tracking with a probabilistic occupancy map. IEEE Transactions on Pattern Analysis and Machine Intelligence 30, 267–282 (2008)
18. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
19. Isard, M., Blake, A.: Condensation-conditional density propagation for visual tracking. International Journal of Computer Vision 29, 5–28 (1998)