Computer Vision and Image Understanding 116 (2012) 1076–1094

Contents lists available at SciVerse ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Fuzzy spatial constraints and ranked partitioned sampling approach for multiple object tracking [☆]

Nicolas Widynski [a,b,*], Séverine Dubuisson [a], Isabelle Bloch [b]

[a] UPMC, CNRS LIP6, Paris, France
[b] Télécom ParisTech, CNRS LTCI, Paris, France

ARTICLE INFO

Article history:
Received 3 August 2011
Accepted 17 July 2012
Available online 25 July 2012

Keywords:
Multiple object tracking
Particle filter
Fuzzy spatial constraints
Partitioned Sampling

ABSTRACT

While particle filters are now widely used for object tracking in videos, the case of multiple object tracking still raises a number of issues. Among them, a first, and very important, problem concerns the exponential increase of the number of particles with the number of objects to be tracked, that can make some practical applications intractable. To achieve good tracking performances, we propose to use a Partitioned Sampling method in the estimation process with an additional feature about the ordering sequence in which the objects are processed. We call it *Ranked Partitioned Sampling*, where the optimal order in which objects should be processed and tracked is estimated jointly with the object state. Another essential point concerns the modeling of possible interactions between objects. As another contribution, we propose to represent these interactions within a formal framework relying on fuzzy sets theory. This allows us to easily model spatial constraints between objects, in a general and formal way. The association of these two contributions was tested on typical videos exhibiting difficult situations such as partial or total occlusions, and appearance or disappearance of objects. We show the benefit of using conjointly these two contributions, in comparison to classical approaches, through multiple object tracking and articulated object tracking experiments on real video sequences. The results show that our approach provides less tracking errors than those obtained with the classical Partitioned Sampling method, without the need for increasing the number of particles.

1. Introduction

Since the 1990's, the particle filter has been widely used in the mono-object tracking community, thanks to its natural dispositions for tracking purposes, its reliability to deal with non-linear systems, and its easiness of implementation. However, extending this methodology to multiple object tracking is not straightforward. Usually estimating more objects needs substantially more particles, the association problem between measures and objects has to be solved, and interactions between objects should be modeled.

The adaptation of particle filters to track several objects has been extensively addressed in the literature, in many different ways. In [1], the authors propose a Jump Markov System (JMS) that models and jointly estimates the object states, the number of objects to track, and the association hypotheses between measures and objects. In [2], the particle filter integrates interactions between objects and measures using a Joint Probabilistic Data Association Filter (JPDAF), that provides an optimal data solution in the Bayesian framework. In [3] the distribution of association hypotheses is computed using a Gibbs sampler. The authors in [4,5] use a Joint Multitarget Probability Density (JMPD) to estimate the number of objects and their states. In [6], the authors model the filtering distribution as a mixture model to handle multiple objects, and use an Adaboost procedure to detect objects leaving and entering the images. Two multi-object algorithms are proposed in [7], namely the Sequential Sampling Particle Filter (SSPF), which individually generates objects using a factorization of the importance weights, and the Independent Partition Particle Filter (IPPF), which considers that the associations between objects and measures are independent over the individual objects.

A major issue with the importance sampling, used in particular in particle filters, is that it suffers from the problem of the curse of dimensionality [8,9]. This means that the particle filter requires a number of particles that increases exponentially with

[☆] This paper has been recommended for acceptance by Y. Aloimonos.

[*] Corresponding author. Address: University of Montreal, Department of Computer Science and Operations Research (DIRO), Montreal, Canada.

E-mail addresses: Nicolas.Widynski@gmail.com (N. Widynski), Severine.Dubuisson@lip6.fr (S. Dubuisson), Isabelle.Bloch@telecom-paristech.fr (I. Bloch).

1077-3142/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cviu.2012.07.002

the number of objects, making the use of a particle filter for multiple object tracking intractable as soon as the number of objects is greater than three. Therefore, the authors in [10,11] proposed a particle filter that avoids this additional cost using a Partitioned Sampling strategy. In [10], they consider an exclusion principle to handle the data association problem (i.e. specifying that a measurement may be associated with at most one object). The principle of Partitioned Sampling is to partition the state space, by considering one element of the partition per object. The objects are processed in a prefixed order, that we call *scenario*, and particles that are the most likely to fit with the real state of the object are selected using a weighted sampling approach. 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 in [12], the use of a prefixed order in the scenario for the joint state estimation can cause a lot of problems, caused by a particle cloud impoverishment effect. To avoid this, in [12], the filtering distribution is modeled as a mixture law, in which each component designates a specific order of processing of the objects, that is estimated using a Partitioned Sampling technique. This idea was also used in [13] to merge different features of an object. However, the number of particles used for each component is fixed, that can limit the performances of the filter if the order of processing is not optimal [12].

Handling several objects in a particle filter raises another type of problem. It is often necessary to model possible interactions between objects, in order to estimate jointly object states. Most approaches make specific hypotheses (see for example [14]), that are directly related to the application, making the model not generalizable. Generally speaking, spatial relations constitute an important type of structural information, useful in scene description and interpretation, as acknowledged in various domains. Although most relations have a clear intuitive meaning, expressing them mathematically is not obvious because they may be vague or imprecise. This is for instance the case for relative directions such as *to the right of*. Modeling such relations in a fuzzy set framework is then appropriate [15] since it allows considering them as a matter of degree, which can be tuned according to the context. For instance saying that two objects are *far* from each other depends on the application context, on the objects, and on the image resolution. Several definitions of fuzzy models of spatial relations have been proposed (see e.g. [16] for a review), and used in structural object recognition and image interpretation [17–19], data clustering [20], or single object tracking [21]. This type of information, to our knowledge, has however not been used so far in a multiple object tracking procedure (spatial constraints have only been used in specific ways, in a non fuzzy formalism [14]), and we show here that it is interesting to do so.

In this paper, that extends a preliminary work in [22], we propose to integrate fuzzy spatial constraints into the particle filter framework for multiple object tracking and to jointly estimate object states and their optimal processing order. We call this approach *Ranked Partitioned Sampling*. This allows us to consider the whole set of possible orders and to automatically prune irrelevant scenarios. We consider here that the number M of objects is known (but can vary over time), and we handle automatically possible hidden parts of objects by other ones.

This article is organized as follows: The particle filter for multiple object tracking is first presented in Section 2. The Partitioned Sampling procedure proposed in [10,11] is described in Section 3. Section 4 presents the fuzzy spatial constraint framework and its introduction in the particle filter framework, as our first contribution. The Ranked Partitioned Sampling, which constitutes the second main contribution of this paper, is then proposed in Section 5, and some considerations about the visibility of objects are given in Section 6. We show experimental results in Section 7, before concluding in Section 8.

2. Particle filter for multiple object tracking

2.1. Particle filter

Let us consider a classical filtering problem and denote by $\mathbf{x}_t \in \mathscr{X}$ the hidden state at time t of an object to be tracked, and by $\mathbf{y}_t \in \mathscr{Y}$ the measurement state extracted from the image sequence. The system describing the temporal evolution of $\mathbf{x}_t$ and the measurement equation are defined as follows:

$$\mathbf{x}_t = f_t(\mathbf{x}_{t-1}, \mathbf{v}_t) \quad (1)$$
$$\mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{w}_t) \quad (2)$$

where f_t models the non-linear temporal evolution of $\mathbf{x}_t$, h_t is the non-linear measurement equation, and $\mathbf{v}_t$ and $\mathbf{w}_t$ are independent white noises. The non-linear Bayesian tracking consists in estimating the *posterior* filtering density function $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ (where $\mathbf{y}_{1:t}$ denotes the series of measures from time 1 to time t), expressed by:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)\; p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int_{\mathscr{X}} p(\mathbf{y}_t|\mathbf{x}'_t)\; p(\mathbf{x}'_t|\mathbf{y}_{1:t-1}) d\mathbf{x}'_t} \quad (3)$$

with $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$ the prediction (or *prior*) density function defined as:

$$p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int_{\mathscr{X}} p(\mathbf{x}_t|\mathbf{x}_{t-1})\; p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t-1} \quad (4)$$

When the filtering density function cannot be computed in a closed form, i.e. when the system is non-linear and non-Gaussian, particle filters are used to approximate it by a weighted sum of N Dirac masses $\delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$ centered on hypothetic state realizations $\left\{\mathbf{x}_t^{(n)}\right\}_{n=1}^{N}$ of the state $\mathbf{x}_t$, also called particles [23]. Then, the filtering distribution $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$ 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 nth particle and $w_t^{(n)}$ its weight. If an approximation of $\mathbb{P}(d\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$ 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 $\left\{\mathbf{x}_{t-1}^{(n)}, w_{t-1}^{(n)}\right\}_{n=1}^{N}$ using an importance function $q\left(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_t\right)$.
2. The update step then computes new particle weights using the new observation $\mathbf{y}_t$, as:
$$w_t^{(n)} \propto w_{t-1}^{(n)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(n)}) p\left(\mathbf{x}_t^{(n)}|\mathbf{x}_{t-1}^{(n)}\right)}{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_t)}, \text{ such that } \sum_{i=1}^{N} w_t^{(n)} = 1.$$
3. Resampling techniques are employed to avoid particle degeneracy problems, leading for instance to the classical Sequential Importance Resampling (SIR) filter [23].

2.2. Multiple object tracking using particle filter

When dealing with multiple objects, the previous model has to be adapted. The first proposed approach might be the one in [24], and consists in simply applying the SIR filter [23] to $\mathbf{x}_t$ defined as a concatenation of several objects states $\mathbf{x}_t = (\mathbf{x}_t^1, \ldots, \mathbf{x}_t^M)$, with $\mathbf{x}_t^i \in \mathscr{X}^i$ the unknown state of the ith object and M the fixed number of objects. The same process as the one described in Section 2.1 can then be used. However, a problem involved in multiple object tracking is that the likelihood cannot be directly computed. Indeed, the presence of several objects in the images induces several

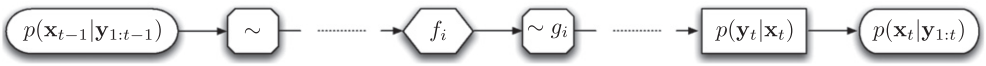

$p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
$\sim$
f_i
$\sim g_i$
$p(\mathbf{y}_t|\mathbf{x}_t)$
$p(\mathbf{x}_t|\mathbf{y}_{1:t})$

Fig. 1. Diagram of the weighted Partitioned Sampling procedure (adapted from [10]).

measures (observations), and the problem is to know which object is related to which measure(s) and to compute its likelihood accordingly. This is known as the data association problem. Let us denote by $\mathbf{y}_t = (\mathbf{y}_t^1, \ldots, \mathbf{y}_t^O)$ the measurement state extracted from the image sequence: it is necessary to determine the good associations between objects $\{\mathbf{x}_t^i\}_{i=1}^M$ and observations $\left\{\mathbf{y}_t^j\right\}_{j=1}^O$. Some usual questions to be answered are: can an object be related to several measures, or at most to one measure? Can a measure be a false alarm? Can it come from several objects? Each possible answer leads to specific hypotheses and to different models of association between data (observations, measures) and targets (objects in image processing). For example, the authors in [3] consider that a measure can come from an object or be a false alarm, and that an object can provide zero, one or several measures. In [2], the authors also consider that a measure can come from an object or be a false alarm, but that an object can provide at most one measure. Other approaches [25] propose to represent the data association problem in a game theoretic formalism, and to find a Nash equilibrium indicating that a satisfactory global configuration was found. Other classical multiple object tracking particle filters have been proposed in [1,4–7].

However, in computer vision, the hypotheses of the data association problem are often relaxed, and the most usual one is that a measure can come from an object or be a false alarm (and one object gives one measure). This hypothesis was used in [10,26], and led to deal with the uncertainty of the origin of measures by only modeling the problem as a visibility of objects. This is simpler, and can be done by using a visibility vector and by estimating states using a particle filter algorithm [10,27]. This implies that the measure process depends on the visibility of objects. In the proposed method, the visibility will be considered too, in particular in the case of hidden objects. This will be detailed in Section 6.

2.3. Problem of state vector dimension

Due to the importance sampling procedure used in particle filters, an increase in the dimensionality of $\mathbf{x}_t$ systematically induces an increase in the variance of the particle weights, leading to a potential impoverishment of the particle set (i.e. only a small part of this set actually represents the object). The issues related to high dimensions have been highlighted in [28]. Filters dedicated to multiple object tracking have been proposed in [29,30], and the Rao-Blackwellised particle filter was proposed in [31]. However, the effective implementation of these filters is not always straightforward and may even not be possible.

In the case of tracking two objects, where $\mathbf{x}_t^i$ and $\mathbf{x}_t^j$ are two states evolving in the same space $\mathscr{X}^*$, it has been shown in [11] that N^2 particles are necessary to achieve the same level of tracking performance as when tracking a single object with N particles. Similarly, it can be shown that for M objects, N is exponentially increasing with the dimension of $\mathbf{x}_t$.

To tackle the dimensionality problem, the authors in [11] 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 process is called Partitioned Sampling, and is presented next.

3. Partitioned Sampling

The Partitioned Sampling (PS), introduced in [10,11], decomposes the joint state space $\mathscr{X}$ into a partition (or a Cartesian product) of M elements: $\mathscr{X} = \mathscr{X}^1 \times \ldots \times \mathscr{X}^M$, with $\mathscr{X}^i$ the state space of the ith object. The construction of this partition is implicit, and consists in associating each element $\mathscr{X}^i$ of the partition with an object i, and for each of them, applying the transition (dynamics) and performing a weighted resampling operation. It is then necessary to define one transition function and one weighted resampling function per object. The partitioning is supposed to be known, and for multiple object tracking, an element could be one object or a group of objects. Here we assume that the partition is defined according to the set of objects to be tracked (one element per object). In the following, we first explain the weighted resampling strategy, and then the global algorithm of Partitioned Sampling.

3.1. Weighted resampling

A weighted resampling operation transforms a particle set $\left\{\mathbf{x}_t^{(n)}, w_t^{(n)}\right\}_{n=1}^N$ into another one $\left\{\tilde{\mathbf{x}}_t^{(n)}, \frac{w_t^{(n)}}{\rho^{(n)}}\right\}_{n=1}^N$ while keeping the distribution intact [10,32]. We consider a strictly positive function g, also called *weighting function*. The resampling process aims at resampling the particles set according to the peaks of g, i.e. generating more samples for higher values of g. The weighting factors $\{\rho^{(n)}\}_{n=1}^N$ are called *importance weights* and are defined as $\rho^{(n)} = g\left(\mathbf{x}_t^{(n)}\right) / \sum_{u=1}^N g\left(\mathbf{x}_t^{(u)}\right)$. Finally the particle set $\left\{\tilde{\mathbf{x}}_t^{(n)}, \frac{w_t^{(n)}}{\rho^{(n)}}\right\}_{n=1}^N$ is obtained by simulating according to the distribution defined by the weights $\{\rho^{(u)}\}_{u=1}^N$, i.e. $\tilde{\mathbf{x}}_t^{(n)} \sim \sum_{u=1}^N \rho^{(u)} \delta_{\mathbf{x}_t^{(u)}}(\mathbf{x}_t)$.

3.2. The Partitioned Sampling procedure

By denoting $\sim$ the resampling procedure according to the particle weights $\left\{w_t^{(n)}\right\}_{n=1}^N$, f_i the dynamics process of object i, possibly conditioned by objects already generated $\mathbf{x}_t^{1:i-1} \triangleq \{\mathbf{x}_t^k\}_{k=1}^{i-1}$, and $\sim g_i$ the weighted resampling operation of the ith object, the weighted Partitioned Sampling operation is summarized in Fig. 1, and the whole algorithm is given in Algorithm 1.

Algorithm 1. Approximation of the posterior distribution $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$ by a particle filter algorithm using Partitioned Sampling with M objects (see text for notations).

```
Input: Approximation of the posterior distribution at $t-1$:
    $\mathbb{P}(d\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \simeq \sum_{n=1}^N w_{t-1}^{(n)} \delta_{\mathbf{x}_{t-1}^{(n)}}(d\mathbf{x}_{t-1})$
Output: Approximation $P_N$ of the posterior distribution at $t$:
    $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$
begin
 1. Resample $\left\{\mathbf{x}_{0:t-1}^{(n)}, 1/N\right\}_{n=1}^N$ from $\left\{\mathbf{x}_{0:t-1}^{(n)}, w_{t-1}^{(n)}\right\}_{n=1}^N$ using the
    multinomial resampling algorithm [33,34].
 2. Partitioned Sampling: for $i = 1, \ldots, M-1$ do
    • Generate the object of index $i$: for $n = 1, \ldots, N$ do
```

```
        $\bar{\mathbf{x}}_t^{i,(n)} \sim q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^{i,(n)}, y_t)$
    • Compute the resampling probabilities $\{\rho^{(n)}\}_{n=1}^N$: for $n = 1, \ldots, N$ do
        $\rho^{(n)} \propto \frac{g_i(\bar{\mathbf{x}}_t^{i,(n)})}{q(\bar{\mathbf{x}}_t^{i,(n)}|\mathbf{x}_{t-1}^{i,(n)}, y_t)}$,   such that $\sum_{u=1}^N \rho^{(u)} = 1$
    • Resample using the weighted resampling: for $n = 1, \ldots, N$ do
        ○ Select the index $k^{(n)}$ of a sample randomly generated from probabilities $\{\rho^{(u)}\}_{u=1}^N$:
            $k^{(n)} \sim p\left(k|\{\rho^{(u)}\}_{u=1}^N\right) = \sum_{u=1}^N \rho^{(u)} \delta_u^k$
        ○ Set the state: $\tilde{\mathbf{x}}_t^{(n)} = \bar{\mathbf{x}}_t^{(k^{(n)})}$
        ○ Compute the normalized importance weight:
            $\tilde{w}_t^{(n)} \propto \frac{w_t^{(k^{(n)})}}{\rho^{(k^{(n)})}}$,   such that $\sum_{u=1}^N \tilde{w}_t^{(u)} = 1$
    • Copy the particle cloud $\left\{\left(\mathbf{x}_{0:t}^{1:i,(n)}, \mathbf{x}_{0:t-1}^{i+1:M,(n)}\right), w_t^{(n)}\right\}_{n=1}^N = \left\{\left(\mathbf{x}_{0:t}^{1:i-1,(n)}, \tilde{\mathbf{x}}_{0:t}^{i,(n)}, \mathbf{x}_{0:t-1}^{i+1:M,(n)}\right), \tilde{w}_t^{(n)}\right\}_{n=1}^N$
3. for $n = 1, \ldots, N$ do
    • Generate for the last object of index $M$:
        $\mathbf{x}_t^{M,(n)} \sim q(\mathbf{x}_t^M|\mathbf{x}_{t-1}^{M,(n)}, y_t)$
    • Compute importance weights using the multiple object joint likelihood:
        $w_t^{*(n)} \propto w_t^{(n)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(n)}) \prod_{i=1}^M p(\mathbf{x}_t^{i,(n)}|\mathbf{x}_{t-1}^{i,(n)})}{q(\mathbf{x}_t^{M,(n)}|\mathbf{x}_{t-1}^{M,(n)}, y_t)}$,
        such that $\sum_{u=1}^N w_t^{*(u)} = 1$
return $P_N(d\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{n=1}^N w_t^{*(n)} \delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$
```

Although any weighting function g_i asymptotically keeps the posterior probability unchanged, the objective of this step is to obtain an accurate representation of this probability. We consider a factorization of the likelihood such that it allows us to deal with each object independently, hence each observation is related to one object. Let $\mathbf{y}_t = (\mathbf{y}_t^1, \ldots, \mathbf{y}_t^M)$ be the observation vector. The likelihood is given by $p(\mathbf{y}_t|\mathbf{x}_t) = \prod_{i=1}^M p(\mathbf{y}_t^i|\mathbf{x}_t^i)$. Then, the likelihood $h_i = p(\mathbf{y}_t^i|\mathbf{x}_t^i)$ of object i appears to be a natural choice for defining the weighting function g_i, and leads to the diagram proposed in Fig. 2, in which the chosen prior importance distribution for the dynamics is $f_i = p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$. Note that the assumption of independence of the observations conditionally to the states is commonly used for modeling the likelihood in a simple way.

3.3. 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 [12], the order of processing 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. Illustrations of this drawback can be found in [35] (Chapter 4).

The method presents an additional difficulty. If occlusions occur, one may quite rightly handle visible objects first, and hence adopt a dynamic ordering strategy. The solution proposed in [10] is called *Branched Partitioned Sampling* (BPS), and consists in adding a vector of visibility to the global estimation, and then in recursively grouping together particles with an identical realization of this vector, generating an hypothesis tree. Considering a tracking problem with M objects, this method may divide particles into M! hypotheses. This solution has however two major drawbacks. First, by dividing N particles into M! hypotheses, the interest of the Partitioned Sampling is lost, since the particles no longer try to survive over a large set of N particles but over possible irrelevant sets of N/M! elements. Secondly, the algorithm can propagate particles into sub-trees of the hypothesis tree in which the visibility vector is erroneous, with no possibility of comparing them with the particles of the other sub-trees: this can lead to a false joint posterior distribution estimation.

The *Dynamic Partitioned Sampling* (DPS), proposed in [12], 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 predefined permutation sets, each one owning N/M particles. This strategy improves the 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 as the Branched Partitioned Sampling, since particles evolve into subsets.

To overcome these problems, we propose a new sampling strategy, called *Ranked Partitioned Sampling* (RPS). Moreover, in order to take into account interactions between objects in their joint state estimation, we propose to introduce spatial relations in the definition of the probability of an object's state at time t, conditionally to the object's state at $t-1$ and to the other objects' states. We first describe our modeling of spatial relations in Section 4, and its introduction into Partitioned Sampling, then present our Ranked Partitioned Sampling in Section 5.

4. Modeling fuzzy spatial constraints between objects

In this section, we propose a new model for integrating spatial constraints expressed in a fuzzy form in a probabilistic tracking approach.

4.1. Modeling fuzzy spatial constraints

We propose to model explicitly interactions between objects as fuzzy spatial relations defined for one, two or more objects. They will be considered as constraints the objects should satisfy during the tracking process, and are therefore called fuzzy spatial constraints.

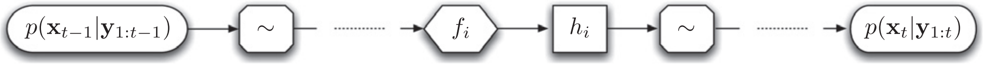

$p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})$
$\sim$
f_i
h_i
$\sim$
$p(\mathbf{x}_t|\mathbf{y}_{1:t})$

Fig. 2. Diagram of the Partitioned Sampling procedure using the likelihood $h_i = p(\mathbf{y}_t^i|\mathbf{x}_t^i)$ as weighting function (adapted from [10]).

Each type of relation is considered as a linguistic variable, taking a small number of linguistic values [36]. 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. For instance for a distance relation, this domain is $\mathbb{R}^+$. If two objects have to be *close* to each other, then a fuzzy set μ^{close} is defined on this domain. For an instance of these two objects, their actual distance d is computed, and $\mu^{close}(d)$ provides the degree to which the relation is satisfied. This principle is used here for all considered relations. Typically such degrees of satisfaction may be evaluated between an object as seen at instant $t-1$ and the same object at t, or between an object and previously processed objects in the order of processing.

In this work, we consider more specifically fuzzy spatial constraints 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 (*to the right of*, *to the left of*, etc.); by ternary operators, such as the concept of rank in an ordered sequence (*the first of*, *in the middle of*, *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. To merge several spatial relations or constraints, a fuzzy fusion operator is used (see e.g. [37]).

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 in a given order of processing. We now define a fuzzy membership function $\nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*) \in [0,1]$ 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 their fusion $\nu_{\tilde{\mathbf{x}}_t}$ as:

$$\nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*) = \mathop{\Xi}_{k=1}^{K} {}^{*}\nu_{\tilde{\mathbf{x}}_t}^{k}(\mathbf{x}_t^*) \tag{5}$$

where Ξ is a fusion operator, for example a t-norm (fuzzy conjunction) [38], and ${}^{*}\nu_{\tilde{\mathbf{x}}_t}^{k} \in [0,1]$ the membership function representing the kth spatial constraint, that defines the degree to which object $\mathbf{x}_t^*$ satisfies the constraint k, with respect to objects $\tilde{\mathbf{x}}_t$. For example, considering a binary fuzzy relation, ${}^{B}\nu_{\tilde{\mathbf{x}}_t}^{k}$ is defined as:

$${}^{B}\nu_{\tilde{\mathbf{x}}_t}^{k}(\mathbf{x}_t^*) = \mathop{\psi}_{l=1}^{L} {}^{B}\nu_{\tilde{\mathbf{x}}_t^l}^{k}(\mathbf{x}_t^*) \tag{6}$$

with ψ a fusion operator, for example a t-norm, and ${}^{B}\nu_{\tilde{\mathbf{x}}_t^l}^{k}(\mathbf{x}_t^*) \in [0,1]$ the value of the membership function of the kth spatial constraint between the current object and object l (component $\tilde{\mathbf{x}}_t^l$ of $\tilde{\mathbf{x}}_t$).

For example, if $\mathbf{x}_t$ corresponds to the spatial coordinates of an object, then the value *at medium distance* between two objects $\mathbf{x}_t^*$ and $\tilde{\mathbf{x}}_t^l$ can be defined by a trapezoidal function such as:

$${}^{B}\nu_{\tilde{\mathbf{x}}_t^l}^{k}(\mathbf{x}_t^*) = \begin{cases} \frac{\|\mathbf{x}_t^* - \tilde{\mathbf{x}}_t^l\|_2 - a}{b-a} & \text{if } \|\mathbf{x}_t^* - \tilde{\mathbf{x}}_t^l\|_2 \in]a;b[\\ 1 & \text{if } \|\mathbf{x}_t^* - \tilde{\mathbf{x}}_t^l\|_2 \in [b,c] \\ 1 - \frac{\|\mathbf{x}_t^* - \tilde{\mathbf{x}}_t^l\|_2 - c}{d-c} & \text{if } \|\mathbf{x}_t^* - \tilde{\mathbf{x}}_t^l\|_2 \in]c;d[\\ 0 & \text{otherwise} \end{cases}$$

with a, b, c, d the parameters of the trapezoidal function with support $[a, d]$ and core $[b, c]$.

The form of function ${}^{B}\nu_{\tilde{\mathbf{x}}_t^l}^{k}$ can be chosen depending on the application. Fig. 3 shows an example of spatial relation constraints for the fingers of a hand. Here we show a simple example using a rectangular membership function (i.e. binary membership) just to illustrate the proposed idea. Section 7 presents more fuzzy spatial relation constraints, with other fuzzy membership functions (see for example Fig. 15). The *middle finger* corresponds to state $\mathbf{x}_t^* = \mathbf{x}_t^m$ for which we want to constrain the central position, and fingers *index finger*, *ring finger* and *little finger* form the concatenated state vector $\tilde{\mathbf{x}}_t = (\tilde{\mathbf{x}}_t^{in}, \tilde{\mathbf{x}}_t^{ri}, \tilde{\mathbf{x}}_t^{li})$ that have already been processed. The binary spatial constraint associated with *index finger* ${}^{B}\nu_{\tilde{\mathbf{x}}_t^{in}}^{k}(\mathbf{x}_t^m)$ represents the degree to which the middle finger is *to the right of* the index finger (Fig. 3a), and the spatial constraints associated with the *ring finger* and *little finger* represent the degree to which the middle finger is *to the left of* the ring and little fingers (Fig. 3b and c). By using the t-norm min as fusion operator ψ (Eq. (6)), we get the global orientation constraint (Fig. 3d). For visualization purpose, in these figures spatial representations of the constraints are used, where gray levels represent the degree of satisfaction of the constraints at each point in space (black corresponds to a high degree of satisfaction). This example will be further developed in Section 7.3.

In a similar way, considering ternary constraints, ${}^{T}\nu_{\tilde{\mathbf{x}}_t}^{k}$ is defined as:

$${}^{T}\nu_{\tilde{\mathbf{x}}_t}^{k}(\mathbf{x}_t^*) = \mathop{\psi}_{l_1=1}^{L} \mathop{\psi}_{l_2=l_1+1}^{L} {}^{T}\nu_{\tilde{\mathbf{x}}_t^{l_1}, \tilde{\mathbf{x}}_t^{l_2}}^{k}(\mathbf{x}_t^*) \tag{7}$$

with ${}^{T}\nu_{\tilde{\mathbf{x}}_t^{l_1}, \tilde{\mathbf{x}}_t^{l_2}}^{k}(\mathbf{x}_t^*) \in [0,1]$ the value of the membership function of the kth spatial constraint between the current object and objects l_1 and l_2.

Lastly, the final membership value $\mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)$ describing fuzzy spatial constraints imposed by a set of objects $\tilde{\mathbf{x}}_t$ is defined from the function $\nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)$ (Eq. (5)), by using a fixed exponent $\gamma \in \mathbb{R}^+$ to control the form of the function $\mu_{\tilde{\mathbf{x}}_t}$:

$$\mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*) = \nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)^{\gamma} \tag{8}$$

Examples of binary, ternary, and quaternary constraints modeled respectively by ${}^{B}\nu_{\tilde{\mathbf{x}}_t^l}^{k}$, ${}^{T}\nu_{\tilde{\mathbf{x}}_t^{l_1}, \tilde{\mathbf{x}}_t^{l_2}}^{k}$ and ${}^{Q}\nu_{\tilde{\mathbf{x}}_t^{l_1}, \tilde{\mathbf{x}}_t^{l_2}, \tilde{\mathbf{x}}_t^{l_3}}^{k}$ will be given in Section 7.

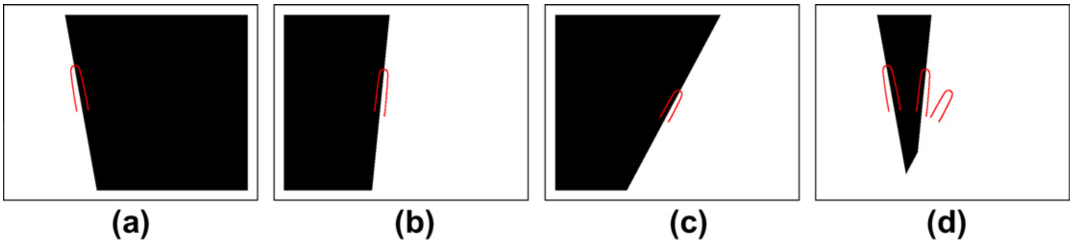


Fig. 3. Spatial representations of orientation constraints between fingers of the hand, expressed for the *middle finger*: (a) relation *to the right of index finger*, (b) relation *to the left of ring finger*, (c) relation *to the left of little finger*, and (d) global constraint result by using the fusion operator ψ = min. Already processed objects (*index finger*, *ring finger* and *little finger*) are displayed in red. Black indicates that the constraint is satisfied. The spatial constraint is computed at each pixel, using the function modeling the relation (here, a rectangular one).

4.2. Introducing fuzzy spatial constraints into a probabilistic framework

The next step consists in introducing the fuzzy spatial constraints $\mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)$ into the probabilistic framework used for tracking. We propose to define the transition probability distribution of an object $\mathbf{x}_t^*$ conditionally to its past realization $\mathbf{x}_{t-1}^*$ and to a group of already processed objects $\tilde{\mathbf{x}}_t$ by combining the transition density $p(\mathbf{x}_t^*|\mathbf{x}_{t-1}^*)$ and spatial constraints imposed by the vector $\tilde{\mathbf{x}}_t$, modeled by $\mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)$.

Introducing fuzzy information in a probabilistic context may be done using the possibility theory [39], or defining fuzzy events [40]. Fuzzy events have already been introduced in a particle filter framework in order to specify the dynamics of parameters [21], and are appropriate for this kind of probabilistic model, since they allow combining in an elegant way fuzzy set semantics and probabilistic ones. In this section, after recalling the concept of fuzzy event, we will employ a similar strategy as in [21] to define the conditional distribution $\mathbb{P}(\mathbf{x}_t^* \in d\mathbf{x}_t^*|\mathbf{x}_{t-1}^*, \tilde{\mathbf{x}}_t)$.

4.2.1. Fuzzy event [40]

A fuzzy event $\mathscr{A}$ is a fuzzy set, defined by a measurable membership function $\mu_{\mathscr{A}} : \mathcal{O} \to [0, 1]$. Under a probability measure $\mathbb{P}$, the probability of a fuzzy event $\mathscr{A}$ is defined as:

$$\mathbb{P}(\mathscr{A}) = \mathbb{E}[\mu_{\mathscr{A}}] = \int_{\mathcal{O}} \mu_{\mathscr{A}}(\mathbf{o})\ d\mathbb{P} \tag{9}$$

The conditional probability of a fuzzy event $\mathscr{A}$ of membership function $\mu_{\mathscr{A}}$ given a fuzzy event $\mathscr{B}$ of membership function $\mu_{\mathscr{B}}$ is expressed as:

$$\mathbb{P}(\mathscr{A}|\mathscr{B}) = \frac{\mathbb{P}(\mathscr{A}\mathscr{B})}{\mathbb{P}(\mathscr{A})} = \frac{\int_{\mathcal{O}} \mu_{\mathscr{A}}(\mathbf{o})\mu_{\mathscr{B}}(\mathbf{o})\ d\mathbb{P}}{\int_{\mathcal{O}} \mu_{\mathscr{A}}(\mathbf{o})\ d\mathbb{P}} \tag{10}$$

where the probability of the conjunction $\mathbb{P}(\mathscr{A}\mathscr{B})$ corresponds to the product t-norm of $\mathscr{A}$ and $\mathscr{B}$ and is defined by the membership function $\mu_{\mathscr{A}}\mu_{\mathscr{B}}$, so as to be consistent with the probabilistic framework.

4.2.2. Fuzzy spatial constraints

We propose to use the concept of fuzzy event to integrate the fuzzy spatial constraints $\mu_{\tilde{\mathbf{x}}_t}$ imposed by a group of already processed objects $\tilde{\mathbf{x}}_t$ into a probabilistic framework, in order to guide the transition of a new object $\mathbf{x}_t^*$. Here the fuzzy event represents the spatial constraints, noted $\mathscr{A}_{\tilde{\mathbf{x}}_t}$, and its membership function is defined by $\mu_{\tilde{\mathbf{x}}_t}$ (Eq. (8)). The transition of the new object $\mathbf{x}_t^*$ should also use its previous state $\mathbf{x}_{t-1}^*$ to take into account its dynamics, which is described by a classical transition model. We define the transition distribution $\mathbb{P}(\mathbf{x}_t^* \in d\mathbf{x}_t^*|\mathbf{x}_{t-1}^*, \tilde{\mathbf{x}}_t)$ depending on the constraints imposed by already processed objects $\tilde{\mathbf{x}}_t$ and on the previous state $\mathbf{x}_{t-1}^*$, with $d\mathbf{x}_t^*$ a fuzzy event of infinitesimal support centered on $\mathbf{x}_t^*$, as:

$$\mathbb{P}(\mathbf{x}_t^* \in d\mathbf{x}_t^*|\mathbf{x}_{t-1}^*, \tilde{\mathbf{x}}_t) = \mathbb{P}(\mathbf{x}_t^* \in d\mathbf{x}_t^*|\mathbf{x}_{t-1}^*, \mathbf{x}_t^* \in \mathscr{A}_{\tilde{\mathbf{x}}_t}) \tag{11}$$

Considering the conditional distribution given in Eq. (10), the transition distribution $\mathbb{P}(\mathbf{x}_t^* \in d\mathbf{x}_t^*|\mathbf{x}_{t-1}^*, \tilde{\mathbf{x}}_t)$ is written as:

$$\begin{aligned}\mathbb{P}(\mathbf{x}_t^* \in d\mathbf{x}_t^*|\mathbf{x}_{t-1}^*, \mathbf{x}_t^* \in \mathscr{A}_{\tilde{\mathbf{x}}_t}) &= \frac{\mathbb{P}(\mathbf{x}_t^* \in d\mathbf{x}_t^*.\mathscr{A}_{\tilde{\mathbf{x}}_t}|\mathbf{x}_{t-1}^*)}{\mathbb{P}(\mathbf{x}_t^* \in \mathscr{A}_{\tilde{\mathbf{x}}_t}|\mathbf{x}_{t-1}^*)} \\ &= \frac{\int_{\mathscr{X}^*} \mu_{d\mathbf{x}_t^*}(\mathbf{x})\ \mu_{\tilde{\mathbf{x}}_t}(\mathbf{x})\ p(\mathbf{x}|\mathbf{x}_{t-1}^*)d\mathbf{x}}{\int_{\mathscr{X}^*} \mu_{\tilde{\mathbf{x}}_t}(\mathbf{x})p(\mathbf{x}|\mathbf{x}_{t-1}^*)d\mathbf{x}} \\ &= D(\mathscr{A}_{\tilde{\mathbf{x}}_t}, \mathbf{x}_{t-1}^*)\ d\mathbf{x}_t^*\ \mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)\ p(\mathbf{x}_t^*|\mathbf{x}_{t-1}^*)\end{aligned} \tag{12}$$

with $\mathscr{X}^*$ the definition space of $\mathbf{x}^*$ and $\mu_{\tilde{\mathbf{x}}_t}$ the fuzzy spatial constraints imposed by $\tilde{\mathbf{x}}_t$ and defined by Eq. (8). The last line in Eq. (12) is obtained by using the fact that the fuzzy set $d\mathbf{x}_t^*$ is of infinitesimal support centered on $\mathbf{x}_t^*$. Generally, the computation of the denominator $D(\mathscr{A}_{\tilde{\mathbf{x}}_t}, \mathbf{x}_{t-1}^*)$ is intractable in a closed form. We propose to approximate this term using a simple S-weighted sampling procedure:

$$D(\mathscr{A}_{\tilde{\mathbf{x}}_t}, \mathbf{x}_{t-1}^*) \approx \left[\sum_{s=1}^{S} \mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}^{(s)})\right]^{-1} \quad \text{with} \quad \left\{\mathbf{x}^{(s)}\right\}_{s=1}^{S} \sim p(\mathbf{x}|\mathbf{x}_{t-1}^*) \tag{13}$$

In practice, only a small number of samples is required (e.g. $S = 10$). Moreover, this computational cost may be greatly reduced thanks to the resampling procedure used in the particle filter: first, it allows computing the term just for distinct particles; secondly, particles with the same ancestors may be used to estimate this term, hence reducing the number S of samples by the number of duplicated particles.

In general, it is difficult to analytically compute $\mu_{\tilde{\mathbf{x}}_t}$, making the simulation according to $\mathbb{P}(\mathbf{x}_t^* \in d\mathbf{x}_t^*|\mathbf{x}_{t-1}^*, \tilde{\mathbf{x}}_t)$ unfeasible. The term $\mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)$ is then considered in the correction phase (i.e. when updating the particle weights), and the simulation of the particles is done by only using $p(\mathbf{x}_t^*|\mathbf{x}_{t-1}^*)$. In cases where a closed form of the function $\mu_{\tilde{\mathbf{x}}_t}$ is known, the simulation according to Eq. (12) would remain difficult (except for very simple cases), but would allow us to generate particles depending on constraints $\mu_{\tilde{\mathbf{x}}_t}$, and then to take into account the transition distribution $p(\mathbf{x}_t^*|\mathbf{x}_{t-1}^*)$ during the weight update step.

4.3. Spatial relations and Partitioned Sampling

Let us now come back to the Partitioned Sampling approach. The dependence between states is handled by our model introducing fuzzy spatial constraints, and we introduce the interaction density function defined in Eq. (8) into the dynamical model, leading to $f_i = p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)\ D(\mathscr{A}_{\mathbf{x}_t^{1:i-1}}, \mathbf{x}_{t-1}^i)\ \mu_{\mathbf{x}_t^{1:i-1}}(\mathbf{x}_t^i)$. This model can be viewed as the pairwise Markov Random Field prior probability used in [13,14,41]. However, in a more general perspective, it is often impossible to directly generate samples from $\mu_{\mathbf{x}_t^{1:i-1}}(\mathbf{x}_t^i)$. Then, we consider $f_i = p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$ whereas the likelihood integrates the interaction term, i.e., $h_i = p_i(\mathbf{y}_t|\mathbf{x}_t^i)\ D(\mathscr{A}_{\mathbf{x}_t^{1:i-1}}, \mathbf{x}_{t-1}^i)\ \mu_{\mathbf{x}_t^{1:i-1}}(\mathbf{x}_t^i)$. This procedure has no impact on the posterior probability since it can be seen as an importance sampling step [14]. Note that a more sophisticated importance distribution could also be used, as in Algorithm 1. Under these assumptions, the approximation process is simplified, and given in Algorithm 2.

Algorithm 2. Approximation of the posterior distribution $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$ by a particle filter algorithm using Partitioned Sampling, with M objects, and using as weighting function for the ith object its likelihood, integrating spatial constraints.

```
Input: Approximation of posterior density at $t-1$:
  $\mathbb{P}(d\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \simeq \sum_{n=1}^{N} w_{t-1}^{(n)} \delta_{\mathbf{x}_{t-1}^{(n)}}(d\mathbf{x}_{t-1})$
Output: Approximation $P_N$ of the posterior density at $t$:
  $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$
begin
 1. Copy the particle cloud $\left\{\mathbf{x}_t^{(n)}, w_t^{(n)}\right\}_{n=1}^{N} = \left\{\mathbf{x}_{t-1}^{(n)}, w_{t-1}^{(n)}\right\}_{n=1}^{N}$
 2. for $i = 1, \ldots, M$ do
    • Resample $\left\{\mathbf{x}_{0:t}^{(n)}, 1/N\right\}_{n=1}^{N}$ from $\left\{\mathbf{x}_{0:t}^{(n)}, w_t^{(n)}\right\}_{n=1}^{N}$ using the
      multinomial resampling algorithm.
    • for $n = 1, \ldots, N$ do
       ○ Generate $\mathbf{x}_t^{i,(n)} \sim p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^{i,(n)})$
       ○ Compute importance weights:
    $w_t^{(n)} \propto p(\mathbf{y}_t^i|\mathbf{x}_t^{i,(n)})\ D(\mathscr{A}_{\mathbf{x}_t^{1:i-1,(n)}}, \mathbf{x}_{t-1}^{i,(n)})\ \mu_{\mathbf{x}_t^{1:i-1,(n)}}(\mathbf{x}_t^{i,(n)})$,
    such that $\sum_{u=1}^{N} w_t^{(u)} = 1$

return $P_N(d\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{n=1}^{N} w_t^{(n)} \delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$
```

5. Ranked Partitioned Sampling (RPS)

The main idea behind the method we propose is to jointly estimate the state of the objects and their order of processing. Objects with highest confidence are considered in earlier stages, because they are supposed to better get through the impoverishment phenomenon of the particles. Each processing order is then compared with the others, implicitly pruning unlikely branches. The adaptive choice of the order of processing aims at limiting the impoverishment effect.

A key point is to introduce a random variable $\mathbf{o}_t^i$ representing the position (or rank) of object i in a processing order. Let $\mathbf{o}_t = (\mathbf{o}_t^1, \ldots, \mathbf{o}_t^M)$ be the set of ranks for all objects at time t, i.e. a permutation over M objects. We call *scenario* the inverse permutation $\mathbf{o}_t^{-1}$ considered at a particular instant t, denoted by $\mathbf{s}_t = (\mathbf{s}_t^1, \ldots, \mathbf{s}_t^M)$, and which represents the ordered sequence of objects. Hence the kth component of a scenario is defined as follows: $\mathbf{s}_t^k = i$ if and only if $\mathbf{o}_t^i = k$ (or in an equivalent manner, $\mathbf{s}_t^k \triangleq \sum_{i=1}^M i\ \delta_{\mathbf{o}_t^i}^k$, with δ_a^b the Kronecker function that equals 1 if $a = b$, 0 otherwise). It indicates the object processed at step k.

Let us consider the following example: four objects are considered, with their ranks $\mathbf{o} = (2, 3, 4, 1)$. This means that object *1* is considered in 2nd position, object 2 in 3rd position, etc. The scenario then corresponds to $\mathbf{s} = (4, 1, 2, 3)$ and indicates in which order objects will be considered in the sequential estimation process (first object 4, then object 1, etc.).

We consider fixed probabilities of transition of the positions (i.e. independent of objects i and time t):

$$\forall i \in \{1, \ldots, M\}, \forall t,\ \mathbb{P}(\mathbf{s}_t^h = i | \mathbf{s}_{t-1}^k = i) \triangleq \mathbb{P}(\mathbf{o}_t^i = h | \mathbf{o}_{t-1}^i = k) \triangleq \alpha_{k,h}. \tag{14}$$

The probability $\mathbb{P}(\mathbf{o}_t^i = h | \mathbf{o}_{t-1}^i = k)$ is meaningful and useful to indicate the probability for object i to move from position k in the scenario at $t-1$ to position h in the scenario at t. On the other hand, $\mathbf{s}_t^k$ is mainly used for indicating the rank of each object at time t, but we are not directly considering probabilities such as $\mathbb{P}(\mathbf{s}_t^k = i | \mathbf{s}_{t-1}^k = j)$ (i.e. probability that object j has the rank k at $t-1$ and is then replaced by object i at this position at t), since in general no prior information is available to derive this probability in a meaningful way. So it is more relevant to deal with $\mathbb{P}(\mathbf{o}_t^i = h | \mathbf{o}_{t-1}^i = k)$.

The use of a predefined and fixed transition matrix allows to intuitively set transition probabilities of the discrete process. For this modeling, we favor static states, and coefficients $\alpha_{k,h}$ have therefore maximal values on the diagonal. However the matrix does not need to be symmetric, as it will be the case in Section 7 (see for example Eq. (21)). This matrix is denoted by $Q_\alpha = (\alpha_{k,h})_{(k,h) \in \{1,\ldots,M\}^2}$.

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

$$\mathbb{P}(\mathbf{o}_t | \mathbf{o}_{t-1}) = \mathbb{P}\left(\mathbf{o}_t^{\mathbf{s}^1} | \mathbf{o}_{t-1}^{\mathbf{s}^1}\right) \prod_{k=2}^{M} \mathbb{P}\left(\mathbf{o}_t^{\mathbf{s}^k} | \mathbf{o}_{t-1}^{\mathbf{s}^k} \triangleq k, \mathbf{o}_t^{\mathbf{s}^1}, \ldots, \mathbf{o}_t^{\mathbf{s}^{k-1}}\right) \tag{15}$$

with $\mathbf{s}^k \triangleq \mathbf{s}_{t-1}^k$, the time subscript being omitted to simplify notations.

In the previous example, let us assume that the ordering $\mathbf{o} = (2, 3, 4, 1)$ was estimated at $t-1$. Then Eq. (15) gives:

$$\mathbb{P}(\mathbf{o}_t | \mathbf{o}_{t-1}) = \mathbb{P}(\mathbf{o}_t^4 | \mathbf{o}_{t-1}^4)\ \mathbb{P}(\mathbf{o}_t^1 | \mathbf{o}_{t-1}^1, \mathbf{o}_t^4)\ \mathbb{P}(\mathbf{o}_t^2 | \mathbf{o}_{t-1}^2, \mathbf{o}_t^4, \mathbf{o}_t^1)\ \mathbb{P}(\mathbf{o}_t^3 | \mathbf{o}_{t-1}^3, \mathbf{o}_t^4, \mathbf{o}_t^1, \mathbf{o}_t^2)$$

The way the joint distribution $\mathbb{P}(\mathbf{o}_t | \mathbf{o}_{t-1})$ is decomposed is thus provided by the estimation at $t-1$ of the scenario $\mathbf{s}_{t-1} = (4, 1, 2, 3)$.

The last conditional distribution in Eq. (15) depends on the transition matrix defined in Eq. (14) and on the positions already allocated to previous objects:

$$\begin{aligned}
&\mathbb{P}\left(\mathbf{o}_t^{\mathbf{s}^k} = h | \mathbf{o}_{t-1}^{\mathbf{s}^k} \triangleq k, \mathbf{o}_t^{\mathbf{s}^1}, \ldots, \mathbf{o}_t^{\mathbf{s}^{k-1}}\right) \\
&\quad = \left[1 - \sum_{j=1}^{k-1} \delta_{\mathbf{o}_t^{\mathbf{s}^j}}^h\right] \left[\alpha_{k,h} + \frac{1}{M-k+1} \sum_{j=1}^{k-1} \alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}\right]
\end{aligned} \tag{16}$$

with $\mathbf{s}^k \triangleq \mathbf{s}_{t-1}^k$. The first term in the product ensures that the probability is set to 0 if the position h has already been assigned. The second term contains the transition probabilities from positions h to k, as defined in Eq. (14), and a term that uses probabilities of transition of the assigned positions (i.e. the positions $\mathbf{o}_t^{\mathbf{s}^1}, \ldots, \mathbf{o}_t^{\mathbf{s}^{k-1}}$ of already processed objects) to balance the distribution in a uniform way. As $k-1$ positions have already been allocated, there remain $M-\ (k-1)$ positions available, that explain the common denominator of the last sum. A proof that Eq. (16) defines a probability distribution is given in Appendix A.

Let us again detail Eq. (16) for the previous example, with $\mathbf{o}_{t-1} = (2, 3, 4, 1)$ and $\mathbf{s}_{t-1} = (4, 1, 2, 3)$. The conditional probabilities defined by Eq. (16) are then:

$$\mathbb{P}(\mathbf{o}_t^4 = h | \mathbf{o}_{t-1}^4 = 1) = \alpha_{1,h}$$

$$\mathbb{P}(\mathbf{o}_t^1 = h | \mathbf{o}_{t-1}^1 = 2, \mathbf{o}_t^4) = \left[1 - \delta_{\mathbf{o}_t^4}^h\right] \left[\alpha_{2,h} + \frac{1}{3} \alpha_{2,\mathbf{o}_t^4}\right]$$

$$\mathbb{P}(\mathbf{o}_t^2 = h | \mathbf{o}_{t-1}^2 = 3, \mathbf{o}_t^4, \mathbf{o}_t^1) = \left[1 - (\delta_{\mathbf{o}_t^4}^h + \delta_{\mathbf{o}_t^1}^h)\right] \left[\alpha_{3,h} + \frac{1}{2}(\alpha_{3,\mathbf{o}_t^4} + \alpha_{3,\mathbf{o}_t^1})\right]$$

$$\mathbb{P}(\mathbf{o}_t^3 = h | \mathbf{o}_{t-1}^3 = 4, \mathbf{o}_t^4, \mathbf{o}_t^1, \mathbf{o}_t^2) = \left[1 - (\delta_{\mathbf{o}_t^4}^h + \delta_{\mathbf{o}_t^1}^h + \delta_{\mathbf{o}_t^2}^h)\right] \left[\alpha_{4,h} + \alpha_{4,\mathbf{o}_t^4} + \alpha_{4,\mathbf{o}_t^1} + \alpha_{4,\mathbf{o}_t^2}\right]$$

Let us now detail the joint estimation of state and scenario. We decompose the joint transition density $p(\mathbf{x}_t,\mathbf{o}_t | \mathbf{x}_{t-1},\mathbf{o}_{t-1})$ in the following way:

$$p(\mathbf{x}_t, \mathbf{o}_t | \mathbf{x}_{t-1}, \mathbf{o}_{t-1}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{o}_t)\ \mathbb{P}(\mathbf{o}_t | \mathbf{o}_{t-1}) \tag{17}$$

where $p(\mathbf{o}_t | \mathbf{o}_{t-1})$ is the transition distribution defined in Eq. (15). Conditioned by the sequence order defined by $\mathbf{o}_t$ and $\mathbf{s}_t$, the transition density of the vector state $\mathbf{x}_t$ is decomposed by first considering the objects placed in the earliest positions:

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{o}_t) \triangleq \prod_{k=1}^{M} p\left(\mathbf{x}_t^{\mathbf{s}_t^k} | \mathbf{x}_{t-1}^{\mathbf{s}_t^k}, \mathbf{x}_t^{\mathbf{s}_t^1}, \ldots, \mathbf{x}_t^{\mathbf{s}_t^{k-1}}\right) \tag{18}$$

Hence, the order of the Markov chain defined by the components of $\mathbf{x}_t$, i.e. by the components of objects $\{\mathbf{x}_t^i\}_{i=1}^M$, is given by the discrete process of the list of ranks $\mathbf{o}_t$, or, in an equivalent manner, by the discrete process representing a scenario $\mathbf{s}_t$.

For the example of scenario $\mathbf{s}_{t-1} = (4, 1, 2, 3)$, if we suppose that its transition at t is $\mathbf{s}_t = (4, 2, 1, 3)$, then $\mathbf{s}_t$ defines the decomposition of the joint probability density of the state dynamics (Eq. (18)):

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{o}_t) = p(\mathbf{x}_t^4 | \mathbf{x}_{t-1}^4)\ p(\mathbf{x}_t^2 | \mathbf{x}_{t-1}^2, \mathbf{x}_t^4)\ p(\mathbf{x}_t^1 | \mathbf{x}_{t-1}^1, \mathbf{x}_t^4, \mathbf{x}_t^2)\ p(\mathbf{x}_t^3 | \mathbf{x}_{t-1}^3, \mathbf{x}_t^4, \mathbf{x}_t^2, \mathbf{x}_t^1)$$

According to the discussion in Section 3, we simplify the dynamic process by conditioning an object's state only by its own state at $t-1$:

$$f_{\mathbf{s}_t^k} = p\left(\mathbf{x}_t^{\mathbf{s}_t^k} | \mathbf{x}_{t-1}^{\mathbf{s}_t^k}, \mathbf{x}_t^{\mathbf{s}_t^1}, \ldots, \mathbf{x}_t^{\mathbf{s}_t^{k-1}}\right) = p\left(\mathbf{x}_t^{\mathbf{s}_t^k} | \mathbf{x}_{t-1}^{\mathbf{s}_t^k}\right) \tag{19}$$

This choice is considered for the simplicity of the presented algorithm. Defining $f_{\mathbf{s}_t^k}$ by an importance function q, such as in Algorithm 1, is of course possible. In the same way, the likelihood is defined as:

$$p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{o}_t) \triangleq \prod_{k=1}^{M} p\left(\mathbf{y}_t^{\mathbf{s}_t^k} | \mathbf{x}_t^{\mathbf{s}_t^k}\right) \tag{20}$$

To summarize, at a time t, for each particle, the algorithm first generates a scenario. Then, at position k of the process (scenario), it resamples the set of particles, proposes a new state of the object $\mathbf{s}_t^k$ at this position using dynamics, and computes the likelihood.

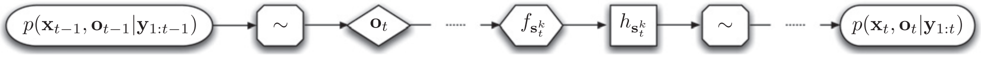

$p(\mathbf{x}_{t-1}, \mathbf{o}_{t-1}|\mathbf{y}_{1:t-1})$
$\sim$
$\mathbf{o}_t$
$f_{\mathbf{s}_t^k}$
$h_{\mathbf{s}_t^k}$
$\sim$
$p(\mathbf{x}_t, \mathbf{o}_t|\mathbf{y}_{1:t})$

Fig. 4. Diagram of the Ranked Partitioned Sampling procedure using the likelihood as weighting function.

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 the final likelihood, taking into account interactions between objects, $h_{\mathbf{s}_t^k} = p\left(\mathbf{y}_t^{\mathbf{s}_t^k}|\mathbf{x}_t^{\mathbf{s}_t^k}\right) D\left(\mathscr{A}_{\mathbf{x}_t^{\mathbf{s}_t^1:\mathbf{s}_t^{k-1}}}, \mathbf{x}_{t-1}^{\mathbf{s}_t^k}\right) \mu_{\mathbf{x}_t^{\mathbf{s}_t^1:\mathbf{s}_t^{k-1}}}\left(\mathbf{x}_t^{\mathbf{s}_t^k}\right)$, we obtain the diagram in Fig. 4, and the approximation process is given in Algorithm 3. As in Algorithm 2, we consider, to simplify, the weighting function $g_i = h_i$ and the importance distribution $f_i = p(\mathbf{x}_t^i|\mathbf{x}_{t-1}^i)$.

Algorithm 3. Approximation of the posterior distribution $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$ by a particle filter algorithm using Ranked Partitioned Sampling, with M objects, using as weighting function for the ith object its likelihood, integrating spatial constraints.

```
Input: Approximation of the posterior distribution at $t-1$:
  $\mathbb{P}(d\mathbf{x}_{t-1}, \mathbf{o}_{t-1}|\mathbf{y}_{1:t-1}) \simeq \sum_{n=1}^N w_{t-1}^{(n)} \delta_{\mathbf{o}_{t-1}^{(n)}}^{\mathbf{o}_{t-1}} \delta_{\mathbf{x}_{t-1}^{(n)}}(d\mathbf{x}_{t-1})$
Output: Approximation $P_N$ of the posterior distribution at $t$:
  $\mathbb{P}(d\mathbf{x}_t, \mathbf{o}_t|\mathbf{y}_{1:t})$
begin
 1. Copy the particle cloud $\left\{\left(\mathbf{x}_t^{(n)}, \mathbf{o}_t^{(n)}\right), w_t^{(n)}\right\}_{n=1}^N = \left\{\left(\mathbf{x}_{t-1}^{(n)}, \mathbf{o}_{t-1}^{(n)}\right), w_{t-1}^{(n)}\right\}_{n=1}^N$
 2. Generate the set of ranks of objects and the scenario: for $n = 1, \ldots, N$ do
    • Generate $\mathbf{o}_t^{(n)} \sim \mathbb{P}(\mathbf{o}_t|\mathbf{o}_{t-1}^{(n)})$ according to Eq. (15)
    • Compute the scenario $\mathbf{s}_t^{(n)}$: $\forall i = 1, \ldots, M,\ \mathbf{s}_t^{i,(n)} = \sum_{k=1}^M k\ \delta_{\mathbf{o}_t^{k,(n)}}^i$
 3. for $i = 1, \ldots, M$ do
    • Resample $\left\{\left(\mathbf{x}_{0:t}^{(n)}, \mathbf{o}_{0:t}^{(n)}\right), 1/N\right\}_{n=1}^N$ from $\left\{\left(\mathbf{x}_{0:t}^{(n)}, \mathbf{o}_{0:t}^{(n)}\right), w_t^{(n)}\right\}_{n=1}^N$ using the multinomial resampling algorithm.
    • for $n = 1, \ldots, N$ do
       ○ Let $s^i \triangleq \mathbf{s}_t^{i,(n)}$
       ○ Generate $\mathbf{x}_t^{s^i,(n)} \sim p\left(\mathbf{x}_t^{s^i}|\mathbf{x}_{t-1}^{s^i,(n)}\right)$
       ○ Compute importance weights:
         $$w_t^{(n)} = p\left(\mathbf{y}_t^{s^i}|\mathbf{x}_t^{s^i,(n)}\right) D\left(\mathscr{A}_{\mathbf{x}_t^{s^1:s^{i-1},(n)}}, \mathbf{x}_{t-1}^{s^i,(n)}\right) \mu_{\mathbf{x}_t^{s^1:s^{i-1},(n)}}\left(\mathbf{x}_t^{s^i,(n)}\right)$$
         $$s.t. \sum_{u=1}^N w_t^{(u)} = 1$$
return $P_N(d\mathbf{x}_t, \mathbf{o}_t|\mathbf{y}_{1:t}) = \sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{o}_t^{(n)}}^{\mathbf{o}_t} \delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$
```

6. Dealing with hidden objects

In a scene containing several objects, some of them may be occluded by other ones. In order to avoid misleading likelihoods, this has to be considered in the tracking process.

There are many ways to deal with hidden objects. A possible choice is to estimate a visibility vector and then conditioning the likelihood [10,27]. The visibility for each object can have two states (visible or not). The model includes the probability of each state, as well as the transition probability from each state to the other. It is also possible to simply consider a mixture likelihood with fixed weights [42]. In [26], the visibility is determined by estimating the 3D position of the object. As it has already been discussed in Section 2.2, this problem may also be handled by considering a data association vector [1–3].

In our approach, inspired by [12], we choose to implicitly deal with the visibility problem, instead of modeling and estimating explicitly probabilities of visibility states and their transitions. The visibility is handled in the likelihood by considering scenarios (i.e. ordered sequence of objects to be tracked) in which an object is assumed to be more visible if it appears earlier in the scenario. Thus the visibility itself is not modeled explicitly, which contrasts with previous approaches. For each object in the scenario, regions of the image occupied by previously processed objects in this scenario are not considered for computing the likelihood of the current object. This guarantees that, given that the previous objects are correctly tracked, only the visible part of the current object will be actually involved in the computations related to this object. This will be experienced in Section 7.1. Note that assuming that a measure can come from at most one object can be false in cases of objects which are close to each other: in such cases, the extracted measures and their association to objects also depend on the resolution of the sensor.

The different approaches do not exclude each other, and it is for instance possible to integrate a vector of visibility into our Ranked Partitioned Sampling, by adapting the distribution for generating scenarios of Eq. (16), and conditioning it by the vector of visibility. Nevertheless, the advantage of an implicit modeling is that it does not require to increase the state vector dimension.

7. Experimental results

We propose three experiments, on various image sequences, that will show:

- the interest of the Rank Partitioned Sampling (RPS), associated with the notion of implicit visibility given by the scenario, by comparing it with the Branched Partitioned Sampling (BPS) (Section 7.1);
- the stability of RPS when all possible scenarios seem to be equivalent (Section 7.2);
- the interest of RPS by comparing it with Partitioned Sampling (PS) (Section 7.3);
- the interest of fuzzy spatial constraints by comparing the results with those obtained without constraints (Sections 7.2 and 7.3).

7.1. People tracking

The goal of this first experiment is to track people moving in an indoor environment (Fig. 5). We consider a public sequence [43], from which we extracted 280 frames where three pedestrians walk and occult each other. Here we consider that their trajectories are independent from each other, hence we do not model any spatial constraint. The primary aim of this experiment is to test the behavior of RPS, in particular the importance of the implicit non visibility, and to compare the results with those obtained with BPS, that is an extension of PS dealing with occlusions between objects (see Section 3.3).

The objects (persons) to track are represented by vertical surrounding rectangles with fixed size and orientation. Let $\mathbf{x}_t^i = (x_t^i, y_t^i)^T$ be the unknown state of object i (a person), with $(x_t^i, y_t^i)^T$ the 2D coordinates of its center. 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 $\mathbf{v}_t^i$ a white Gaussian noise,

Fig. 5. Some frames from the People sequence [43].

with diagonal covariance matrix $\sigma_x^2 = \sigma_y^2 = 4^2$. To assess the stability of the approach, we considered three transition matrices

$$Q_\alpha^1 = \begin{bmatrix} 0.8 & 0.15 & 0.05 \\ 0.1 & 0.8 & 0.1 \\ 0.05 & 0.15 & 0.8 \end{bmatrix}$$
$$Q_\alpha^2 = \begin{bmatrix} 0.5 & 0.35 & 0.15 \\ 0.25 & 0.5 & 0.25 \\ 0.15 & 0.35 & 0.5 \end{bmatrix} \quad Q_\alpha^3 = \begin{bmatrix} 0.1 & 0.2 & 0.7 \\ 0.45 & 0.1 & 0.45 \\ 0.7 & 0.2 & 0.1 \end{bmatrix} \tag{21}$$

Matrix Q_α^1 allows an object to be considered in the same order, from one frame to another. This is an example where the transition matrix is not symmetric. For intermediate ranks $1 < k < M$, it is natural to define equal probabilities for the transition to rank $k - i$ or to rank $k + i$, with $i < \min(k, M - k - 1)$, thus giving an object the same chance to gain or loose ranks in the scenario. However, this is not true for the extreme ranks that have less possibilities to change rank. Therefore we increase their probability to gain (for the last object in the scenario) or loose (for the first object in the scenario) ranks. Matrix Q_α^2 adds flexibility by defining less contrasted coefficients (values on the diagonal are lower that those in Q_α^1 while values off diagonal are higher). This could be useful when occulation statuses change quickly. We expect that this should not alter too much the quality of the results. Matrix Q_α^3 supports an object to shift its processing order. It means that visible objects that are computed first are likely to be considered farther and hence hidden at the following step. This should decrease the performance of the method since it corrupts the order of visibility of the objects.

For the BPS, we model a visibility vector $\mathbf{v}_t = (\mathbf{v}_t^1, \ldots, \mathbf{v}_t^M)$, with $\mathbf{v}_t^i = 1$ meaning that the object i is visible, and 0 otherwise. The transition probability from state *visible* to state *hidden*, $p(\mathbf{v}_t^i = 0|\mathbf{v}_{t-1}^i = 1),\ \forall i = 1, \ldots, M$, is 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 (i.e. the objects which are visible are ranked first in the scenario), hence 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 explicitly since the scenario vector determines the visibility of an object (see Section 6). Then, particles with a correct scenario will be more likely duplicated than the other ones.

The likelihood of an object is defined from the distance between a color model histogram and a candidate histogram as in [44]. However, only the visible part of the object is considered to avoid penalizing hidden or partially hidden objects (parts of objects overlapping with previously processed objects in the scenario are considered as hidden). Hence the visible part of an object corresponds to the region which has no intersection with previously processed objects.

Let $\mathring{a}_t^i$ denote the set of pixels covered by the rectangle surrounding object i at t, defined by its state $\mathbf{x}_t^i, h_i^\bullet = \{h_i^\bullet(u)\}_{u=1}^U$ the learned histogram model for object i (computed in the first frame where the object has been manually segmented), and $h_{BG}^\bullet = \{h_{BG}^\bullet(u)\}_{u=1}^U$ the histogram model of the background (computed in a region of interest manually defined in the first frame), and u the index of the bin of an U-length histogram. The likelihood can be written as:

$$\begin{aligned} p\left(\mathbf{y}_t^{\mathbf{s}_t^k}|\mathbf{x}_t^{\mathbf{s}_t^k}, \mathbf{x}_t^{\mathbf{s}_t^1:\mathbf{s}_t^{k-1}}\right) &= p\left(\mathbf{y}_t^{\mathbf{s}_t^k}|\breve{a}_t^{\mathbf{s}_t^k} = \left\{\mathring{a}_t^{\mathbf{s}_t^k} \setminus \cup_{h=1}^{k-1} \mathring{a}_t^{\mathbf{s}_t^h}\right\}\right) \\ &\propto \exp\left(-\lambda\left(d^2\left[h_{\mathbf{s}_t^k}^\bullet, h\left(\breve{a}_t^{\mathbf{s}_t^k}\right)\right] - d^2\left[h_{BG}^\bullet, h\left(\breve{a}_t^{\mathbf{s}_t^k}\right)\right]\right)\right) \end{aligned} \tag{22}$$

with $\lambda = 30$ empirically fixed, d^2 the Bhattacharyya distance and $\breve{a}_t^{\mathbf{s}_t^k}$ the set of visible pixels of object of index $\mathbf{s}_t^k$, determined by already processed objects of indices $\{\mathbf{s}_t^h\}_{h=1}^{k-1}$. An example of the likelihood is given in Fig. 6. When we wrongly consider that an object is always visible (Fig. 6b), the obtained values at the true position of the object are lowered, because of the presence of another object occlud-

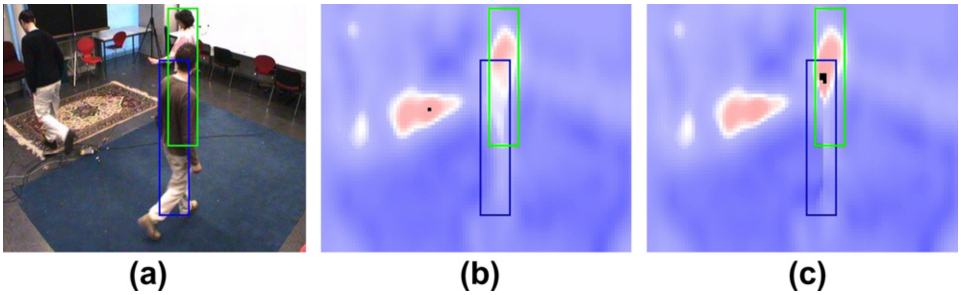


Fig. 6. Tests on the log-likelihood in the case of partial occlusion in the People sequence. (a) Test image, in which we track the person indicated by a green rectangle while it is occluded by the person in the blue one. (b) Likelihood of the person in the green rectangle without using any mask. (c) Likelihood using a mask (pixels belonging to the blue region are not taken into account). Red regions indicate high likelihood values, and black ones correspond to the maximal value of the likelihood.

ing it. Therefore the obtained weights will favor off-centered locations, that actually contain a part of the real object. In Fig. 6c, in which we consider that the blue object is processed first, and thus that it is more visible than the green object, this latter does not take into account pixels belonging to the blue object, which leads to a better localization of the high density values of the likelihood.

We performed the tracking using $N = 500$ particles. Scores are averaged over 20 runs. For Q_α^1, Q_α^2 and Q_α^3, we obtained a mean RMSE of 17.9, 18.7 and 22.4, respectively. With the PS filter, a mean RMSE of 21.6 was obtained. First, differences observed using Q_α^1 and Q_α^2 are very reduced and not really significant. This shows the stability of the method. However, we can observe that experimental results with Q_α^1 are better than the ones with Q_α^2 and Q_α^3. This is consistent with the intuition that visible objects at a certain time are likely to remain visible at the next step. This remark also explains the poor results obtained with Q_α^3, which was expected since its coefficients are not relevant for this application. Finally, the two first matrices led to better results than the PS. Tracking results obtained by the RPS using Q_α^1 and by the PS are illustrated in Fig. 7. Rectangles in red correspond to the estimation of objects (i.e. the Monte Carlo expected value of their center). As mentioned in Section 3.3, 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 differences in the results obtained by the BPS and the RPS (see e.g. second and last images). Fig. 8c and d show the root mean square errors (RMSE) for each tracked person obtained by BPS and RPS, respectively. The error is computed as the root of the average over time of the square difference between the estimated value of the state and its true value. Overall RPS performs better than BPS. Fig. 8b 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. The probabilities estimated for example at times 24, 99, 205, and 212 are consistent with the sequence (Fig. 7). This is observed for instance at times 205, 212 and 259 of Fig. 7b, in which person 3 is hiding person 2, who is hiding person 1. Then, person 3 moves away, when person 2 is still hiding person 1, and this is reflected consistently by the change in probabilities displayed in Fig. 8b, which ranks person 3 first, before favoring person 2 at time 212.

7.2. Ant tracking

In the second experiment, we propose to track ants in a top-view sequence (Fig. 9). This test sequence has been successfully studied in [14], using a MCMC-Based particle filtering approach, and also in a geometry-based particle filter [45]. It contains 750

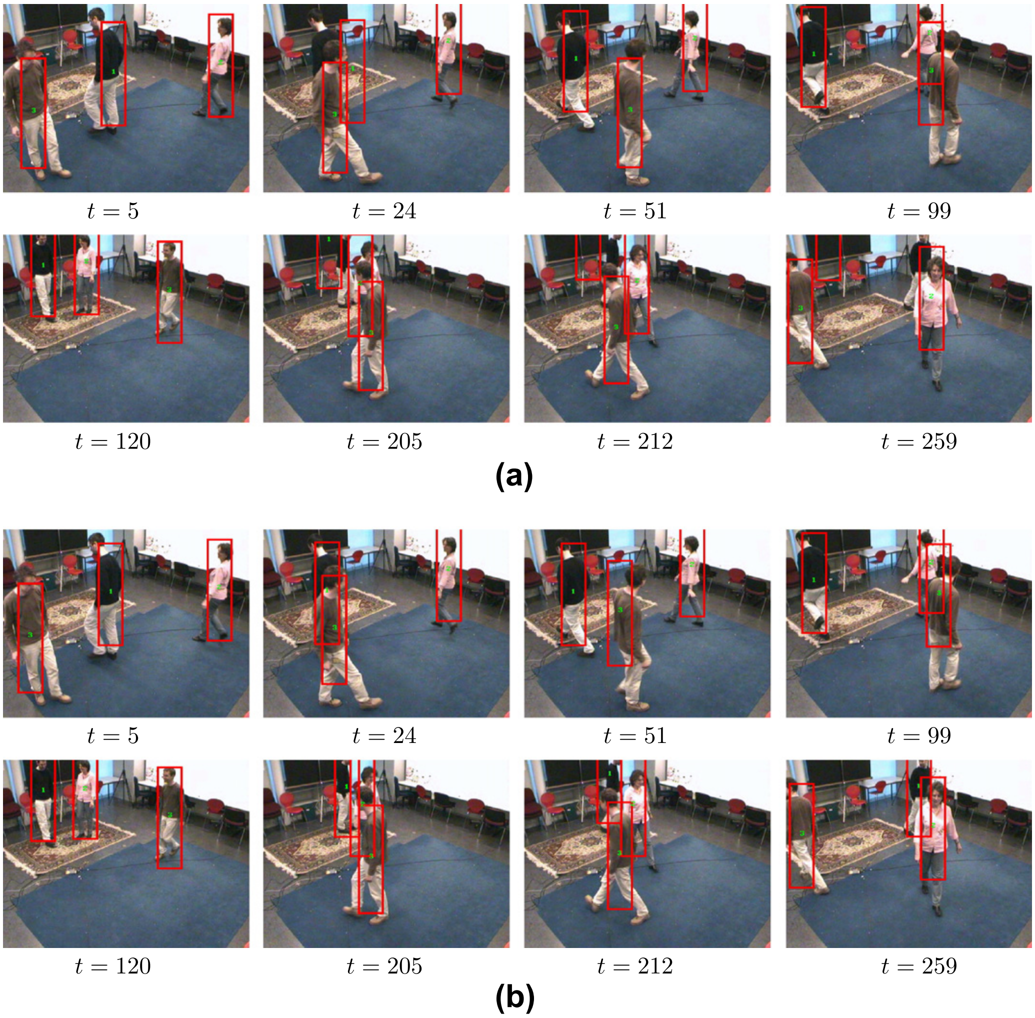


Fig. 7. People tracking results at different times. (a) BPS, (b) RPS. The last one better deals with occlusions, see for example times 24, 212, and 259.

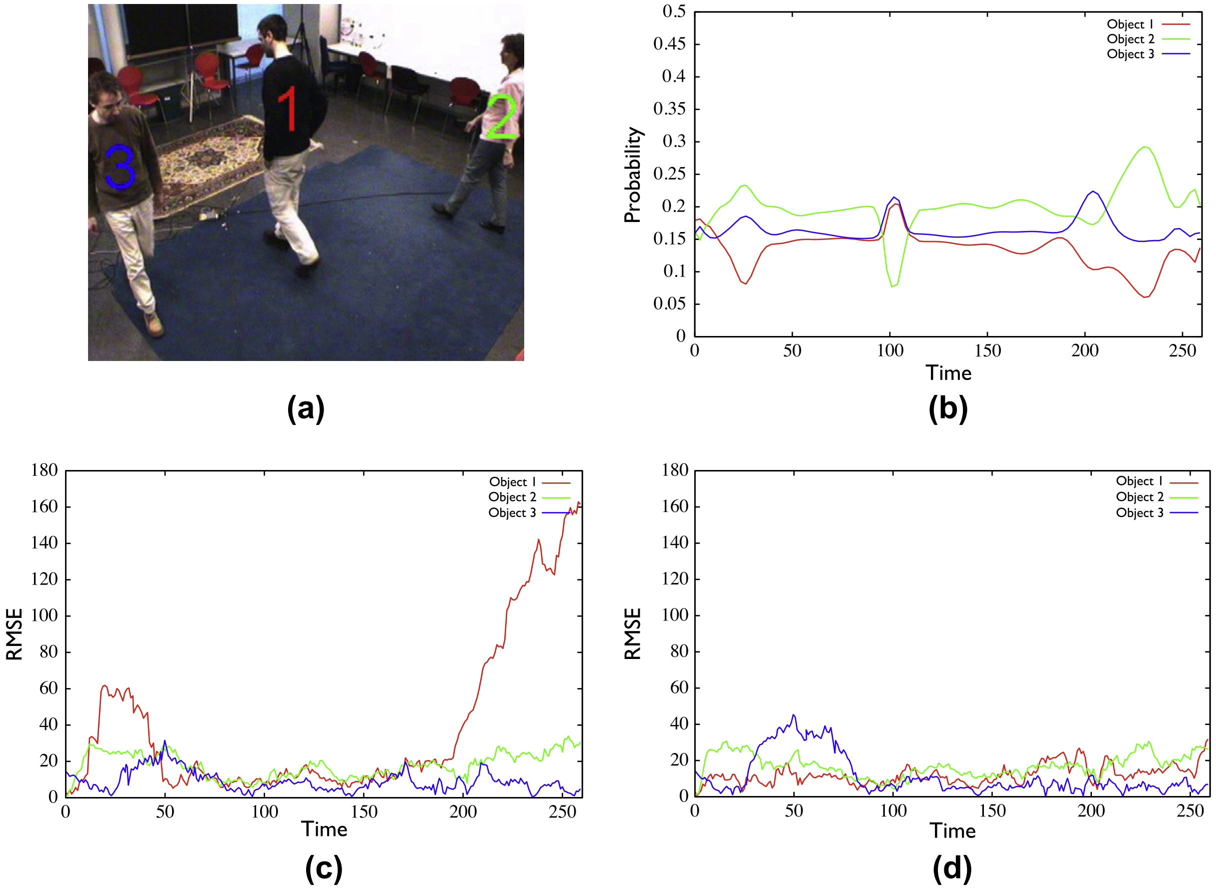


Fig. 8. (a) Indices of the persons present in the scene, (b) posterior probabilities obtained by RPS for a person to be considered first in the processing order, (c) RMSE for BPS and (d) RMSE for RPS (proposed approach).

Fig. 9. Some frames of the Ant sequence [14,45].

frames in which 4–8 ants are moving. Ants are entering and leaving the field of view by a hole in the center of the images, simulating the process of born and death of ants. The number M of ants can then vary over time, but is supposed to be known in each image. Thanks to the position of the camera there is no occlusion, and we exploit this hypothesis by modeling a spatial constraint of exclusion.

As in the first example, objects are represented by rectangles with fixed size (12 × 40 pixels), but their orientation may vary. The state of object i is defined as $\mathbf{x}_t^i = (x_t^i, y_t^i, \theta_t^i)^T$, where $(x_t^i, y_t^i)^T$ denotes the position of the rectangle surrounding the object in the 2D image (center of the rectangle) and θ_t^i its orientation. Dynamics of position and orientation are random walks. The position is subject to a Gaussian white noise, with a diagonal covariance matrix with $\sigma_x^2 = \sigma_y^2 = 11^2$ (those values are slightly higher than those used in [14] since this last method may present a more robust simulation process by using a MCMC method). The noise for orientation is also a Gaussian white one, with variance $\sigma_\theta^2 = 0.3^2$. The transition matrix Q_α is set to:

$$Q_\alpha = \begin{bmatrix} 0.5 & 0.2 & 0.1 & 0.07 & 0.05 & 0.03 & 0.03 & 0.02 \\ 0.15 & 0.43 & 0.15 & 0.1 & 0.07 & 0.05 & 0.03 & 0.02 \\ 0.09 & 0.14 & 0.39 & 0.14 & 0.09 & 0.07 & 0.05 & 0.03 \\ 0.06 & 0.085 & 0.14 & 0.39 & 0.14 & 0.085 & 0.06 & 0.04 \\ 0.04 & 0.06 & 0.085 & 0.14 & 0.39 & 0.14 & 0.085 & 0.06 \\ 0.03 & 0.05 & 0.07 & 0.09 & 0.14 & 0.39 & 0.14 & 0.09 \\ 0.02 & 0.03 & 0.05 & 0.07 & 0.1 & 0.15 & 0.43 & 0.15 \\ 0.02 & 0.03 & 0.03 & 0.05 & 0.07 & 0.1 & 0.2 & 0.5 \end{bmatrix} \quad (23)$$

Probabilities $(\alpha_{k,h})_{(k,h)\in\{1,\ldots,M\}^2}$ have been manually defined, to give flexibility in the rank transitions, and thus to favor the diversification of hypotheses. This is the reason why the diagonal values are not equal: it is natural to consider that ants processed in the middle of the scenario at $t-1$ have a high probability to change their rank, while those processed at the beginning or at the end of the scenario have less chances to change their rank. The precise values of these

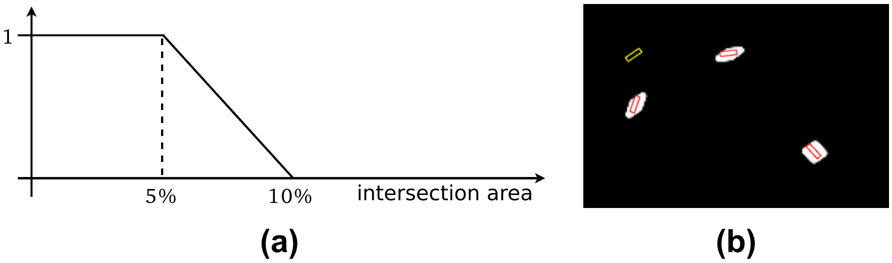


Fig. 10. (a) Decreasing function of the satisfaction degree of the spatial constraint of exclusion ${}^{B}\mu^{exclusion}_{\tilde{\mathbf{x}}^l_t}(\mathbf{x}^m_t)$ depending on the overlap area between $\mathbf{x}^m_t$ and $\tilde{\mathbf{x}}^l_t$. (b) Spatial constraint on the whole image, by considering three ants already processed (displayed in red), and an ant which orientation is superimposed on the constraint image, in yellow. A low gray level indicates that the constraint is satisfied.

probabilities are obtained from the normalization. As the matrix defined in Eq. (21), and for the same reasons, Q_α is not symmetric.

We propose for this experiment to use an exclusion fuzzy spatial constraint, considering that two ants cannot overlap by more than 10% of their own areas, that an overlap of 5% is completely acceptable, and that the degree of satisfaction of the constraint decreases between these two values. This is modeled as a fuzzy set on $\mathbb{R}^+$, where the membership function $\mu^{exclusion}_{\tilde{\mathbf{x}}^l_t}(\mathbf{x}^m_t)$ represents the degree of satisfaction of the constraint as a function of the overlap between $\tilde{\mathbf{x}}^l_t$ and $\mathbf{x}^m_t$ (Fig. 10a). The probability density of the spatial constraint is then defined as proposed in Section 4, from the following membership functions, with only one spatial constraint (hence no Ξ operator is needed here) and with ψ = min in Eq. (6):

$$\mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}^m_t) = \nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}^m_t)^\gamma$$
$$\nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}^m_t) = {}^{B}\nu^{exclusion}_{\tilde{\mathbf{x}}_t}(\mathbf{x}^m_t)$$
$${}^{B}\nu^{exclusion}_{\tilde{\mathbf{x}}_t}(\mathbf{x}^m_t) = \underset{l=1}{\overset{L}{\psi}}\, {}^{B}\mu^{exclusion}_{\tilde{\mathbf{x}}^l_t}(\mathbf{x}^m_t) = \min_{l=1,\ldots,L} {}^{B}\mu^{exclusion}_{\tilde{\mathbf{x}}^l_t}(\mathbf{x}^m_t)$$

where L is the number of objects already processed at t.

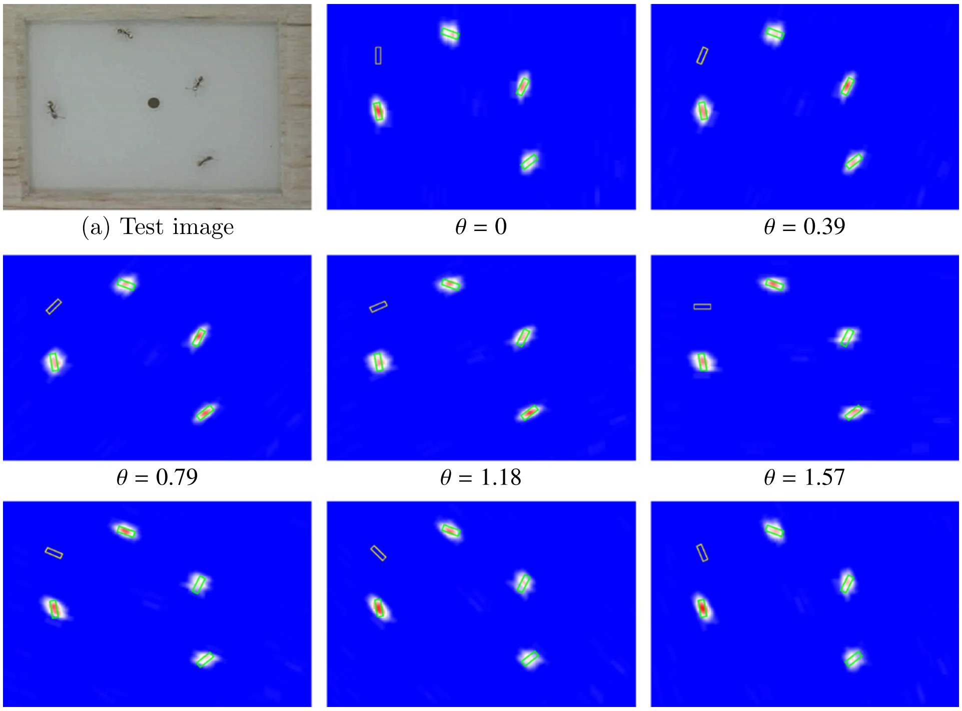
Fig. 10(b) illustrates the spatial constraint of exclusion ${}^{B}\nu^{exclusion}_{\tilde{\mathbf{x}}_t}(\mathbf{x}^m_t)$ computed on the whole image, with respect to the center of an object $i(\mathrm{x}^i_t, \mathrm{y}^i_t)$, by considering that three ants have already been processed.

The likelihood is derived from a simple background subtraction. Let a^i_t be the set of pixels covered by the rectangle of ant i at instant t, I_{BG}:Ω → [0, 255] the gray level background image (image of consecutive frames without ants), I_t: Ω → [0, 255] the gray level image at time t. The likelihood of ant i is written as:

$$p(\mathbf{y}^i_t|\mathbf{x}^i_t) \propto \exp\left(\frac{\lambda}{P_{\mathring{a}^i_t}}\mathrm{card}\left(\mathring{a}^i_t \cap S_{\backslash BG}\right)\right) \qquad (24)$$

with λ = 10 empirically fixed, $P_{\mathring{a}^i_t}$ the number of pixels covered by the set $\mathring{a}^i_t, S_{\backslash BG} = \{p;\ |I_{BG}(p) - I_t(p)| > T,\ p \in \Omega\}$ the set of pixels covered by all the pixels that differ from up to T from the background (here T = 20). The obtained likelihood if shown in Fig. 11, with four ants. As this likelihood is the same for all the ants, an exclusion constraint is necessary to avoid a confusion between them.

Fig. 12 shows results using RPS with and without spatial constraints, and 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 con-



(a) Test image

θ = 0

θ = 0.39

θ = 0.79

θ = 1.18

θ = 1.57

θ = 1.96

θ = 2.36

θ = 2.75

Fig. 11. Tests on the log-likelihood used in Ant sequence. (a) Test image, in which we aim at finding the ant represented in a yellow rectangle in the other images. Obtained results by using the orientation of the yellow rectangle (superimposed on the likelihood) with an angle (in radian) of θ. Red regions correspond to high likelihood values.

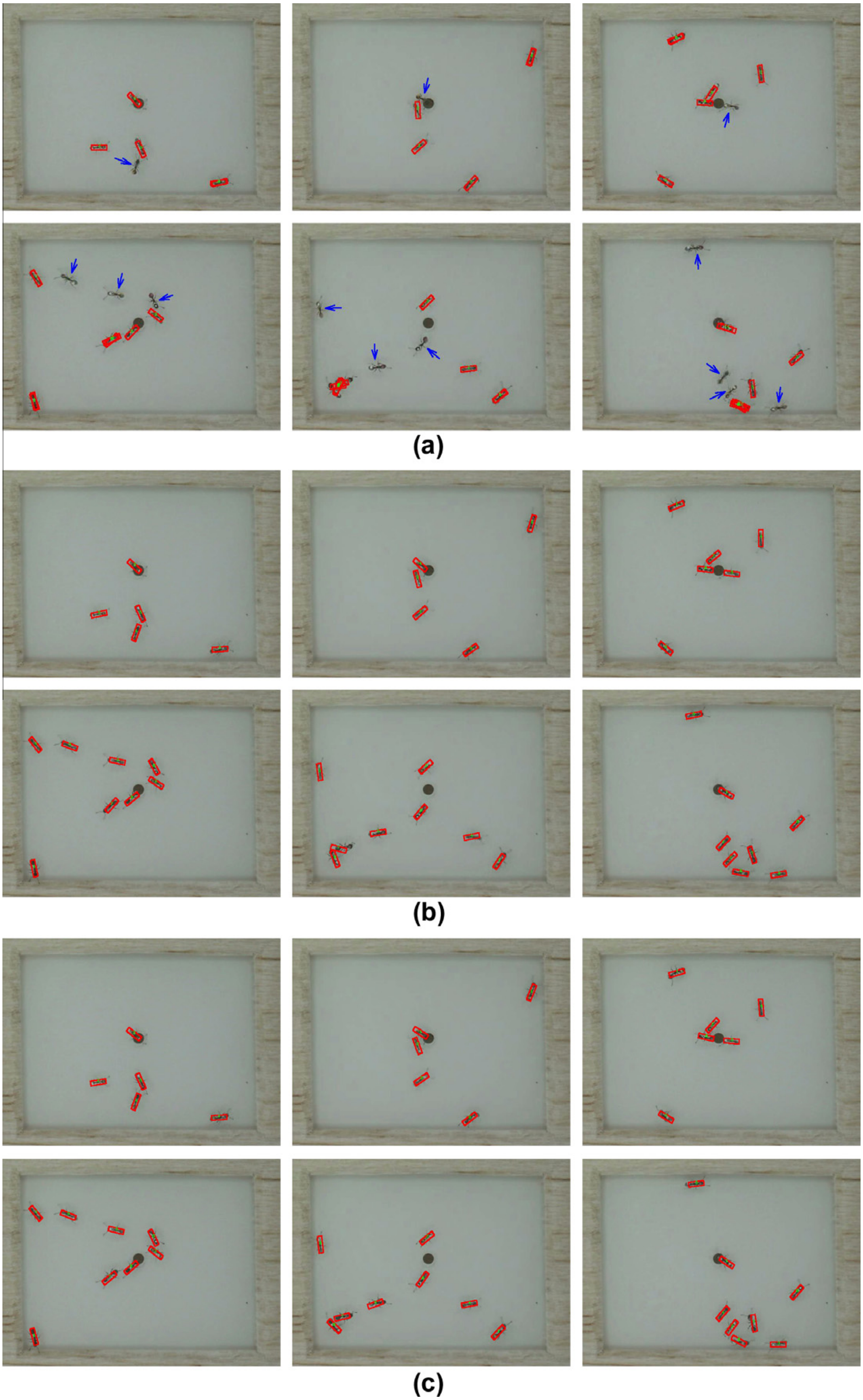


Fig. 12. Ant tracking results. (a) RPS without spatial constraints (blue arrows show ants that are not correctly tracked). (b) PS with spatial constraints. (c) RPS with spatial constraints. All ants are correctly tracked in (b) and (c).

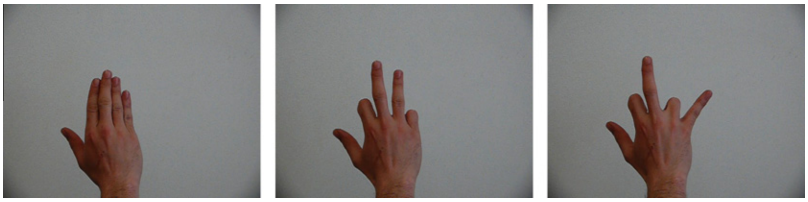


Fig. 13. Frames from hand sequence.

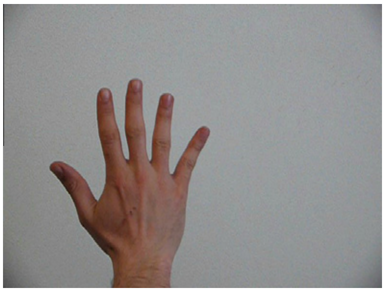


Fig. 14. Image from hand sequence, in which we want to constrain the spatial location and orientation of the *middle finger*, when the other fingers have already been processed and their state estimated.

straint is very clear here (several ants are not tracked without spatial constraints, while they are successfully tracked with such constraints). PS and RPS provide comparable results since all possible processing orders lead to almost identical results. This shows that RPS performs as well as PS when the order does not significantly matter and when the number of objects is small enough to not induce an impoverishment effect.

7.3. Hand tracking

We finally consider a problem of tracking articulated objects (fingers of a hand) on a 800 frame sequence that was acquired to illustrate the main features of the proposed approach. The difficulty of this application is that fingers may be partially or totally hidden (Fig. 13). We propose to track the hand and its fingers by preserving a global consistency of the shape using fuzzy spatial constraints. The global form of the constrained estimation provides a good estimation of the central position and angle of all the fingers, even if they are partially hidden, as will be shown next.

We illustrate the proposed approach on the right hand (this is important to mention because the relative positions of its fingers depend on the considered side). Each finger shape is fixed and represented by a vector of 6 2D-control points, located on the basis of the finger, on the middle and on the fingertips. The fingers are considered as rigid, and thus the control points are fixed. The state of object $i, \mathbf{x}_t^i = (x_t^i, y_t^i, \theta_t^i)^T$, contains the coordinates of its center $(x_t^i, y_t^i)^T$ and its orientation θ_t^i. Dynamics of position and orientation are random walks. A white Gaussian noise model is considered, and a diagonal covariance matrix with variances $\sigma_x^2 = \sigma_y^2 = 8^2$ for position, and $\sigma_\theta^2 = 0.1^2$ for orientation. The transition matrix Q_α is set to:

$$Q_\alpha = \begin{bmatrix} 0.6 & 0.25 & 0.1 & 0.05 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.1 & 0.2 & 0.5 & 0.2 \\ 0.05 & 0.1 & 0.25 & 0.6 \end{bmatrix} \tag{25}$$

Here again, this matrix permits a flexible change in position in the order of processing, and makes the tracking robust to the change of visibility of fingers.

We propose to use fuzzy spatial constraints to overcome this problem and guarantee a good spatial consistency in the tracking results. Although they could be automatically learned, we consider here fixed spatial relations between the fingers. These constraints can be binary relations (distance and orientation), or higher order relations such as alignment. In our experiments, we defined:

- A binary constraint on the relative orientation, denoted by ${}^B\mu^{angle}$: the angle is considered as a linguistic variable that can take here two values, namely *approximately* $-\pi/8$ and *approximately* $\pi/8$ (which are considered as "ideal" values). The semantics of these values are defined by fuzzy sets in the domain of angles, with trapezoidal membership functions centered at $-\pi/8$ and $\pi/8$ respectively, and with a support of length $\pi/4$ (denoted by ${}^B\mu^{-\pi/8}$ and ${}^B\mu^{\pi/8}$). This means that if two neighbor fingers are oriented at $\pi/8$ from each other, the degree of satisfaction of the constraint will be 1, and it will be 0 if they form an angle larger than $3\pi/8$, with decreasing values in between.
- A binary constraint on the distance, denoted by ${}^B\mu^{distance}$. Again two linguistic values are considered: *close to* and *far from*, whose semantics are defined as membership functions ${}^B\mu^{close\ to}$ and ${}^B\mu^{far\ from}$. The parameters of these functions are set according to the distance between fingers in the images.
- One ternary or quaternary (depending on the number of objects already processed) constraint of alignment ${}^Q\mu^{alignment}$: a linear regression is computed from the positions of the base of at least two already processed fingers, and we consider that fingers cannot move away from the regression line more than a fixed distance threshold. This is modeled as a decreasing membership function with respect to the distance to the regressing line.
- One binary constraint of exclusion ${}^B\mu^{exclusion}$, defined as in Section 7.2, with no overlap between fingers larger than 50% allowed. Note that in this experiment, we consider that fingers cannot cross. Otherwise, the exclusion constraint defined here would certainly be too strong. A solution would be to define an interaction constraint that allows object crossing while it forbids overlaps. It could be done using the shape and the orientation of the object.

The operator Ξ involved in the fusion of constraints (Eq. (5)) is defined as the minimum. Its conjunctive behavior allows favoring solutions where all constraints are satisfied. The operator ψ in Eq. (6) is also the minimum. Let us now detail an example of fuzzy spatial constraint formulation expressed for the state of the *middle finger* $\mathbf{x}_t^m$, assuming that the other fingers (denoted by $\mathbf{x}_t^{in}, \mathbf{x}_t^{ri}$ and $\mathbf{x}_t^{li}$ for the *index finger*, the *ring finger* and the *little finger*, respectively) have already been processed in the scenario:

$$\begin{aligned}
\mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^m) &= \nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^m)^\gamma \\
\nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^m) &= \mathop{\Xi}_{k=1}^{K} \nu_{\tilde{\mathbf{x}}_t}^k(\mathbf{x}_t^m) \\
&= \min\left({}^B\nu_{\tilde{\mathbf{x}}_t}^{distance}(\mathbf{x}_t^m), {}^B\nu_{\tilde{\mathbf{x}}_t}^{angle}(\mathbf{x}_t^m), {}^Q\nu_{\tilde{\mathbf{x}}_t}^{alignment}(\mathbf{x}_t^m), {}^B\nu_{\tilde{\mathbf{x}}_t}^{exclusion}(\mathbf{x}_t^m)\right) \\
{}^B\nu_{\tilde{\mathbf{x}}_t}^{distance}(\mathbf{x}_t^m) &= \mathop{\psi}_{l=1}^{L} {}^B\mu_{\tilde{\mathbf{x}}_t^l}^{distance}(\mathbf{x}_t^m) \\
&= \min\left({}^B\mu_{\tilde{\mathbf{x}}_t^{ri}}^{close\ to}(\mathbf{x}_t^m), {}^B\mu_{\tilde{\mathbf{x}}_t^{in}}^{close\ to}(\mathbf{x}_t^m), {}^B\mu_{\tilde{\mathbf{x}}_t^{li}}^{far\ from}(\mathbf{x}_t^m)\right)
\end{aligned}$$

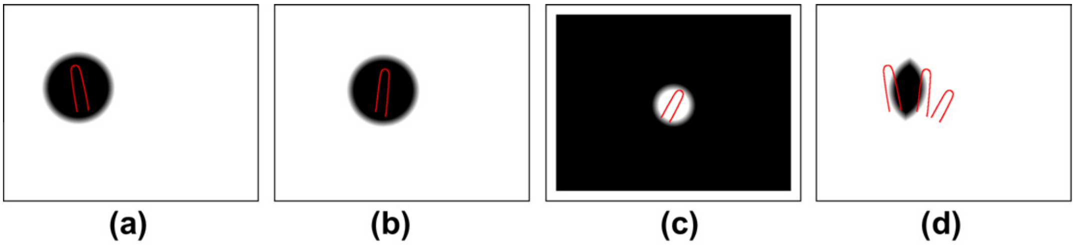


Fig. 15. Fuzzy spatial representation of the distance constraint for the hand, with respect to the *middle finger*. (a) Relation *close to* the *index finger*. (b) Relation *close to* the *ring finger*. (c) Relation *far from* the *little finger*. (d) Global constraint resulting from the conjunctive fusion.

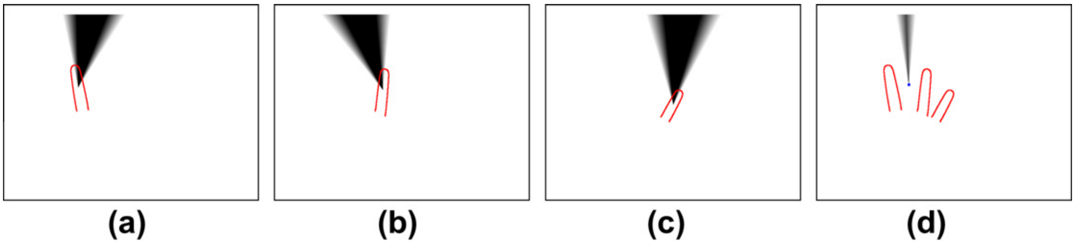


Fig. 16. Fuzzy spatial representation of the orientation constraint with respect to the *middle finger*. (a) Relation *at approximately* $-\pi/8$ of the *index finger*. (b) Relation *at approximately* $\pi/8$ of the *ring finger*. (c) Relation *at approximately* $\pi/8$ of the *little finger*. (d) Global constraint resulting from the conjunctive fusion.

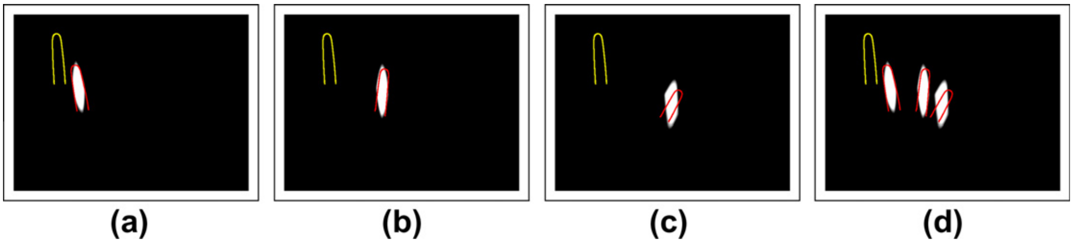


Fig. 17. Fuzzy spatial representation of the exclusion constraint, with respect to the *middle finger*. The finger's orientation is superimposed in yellow on the images of constraints. Exclusion principles seen from (a) *index finger*, (b) *ring finger*, (c) *little finger*. (d) Global exclusion constraint after fusion.

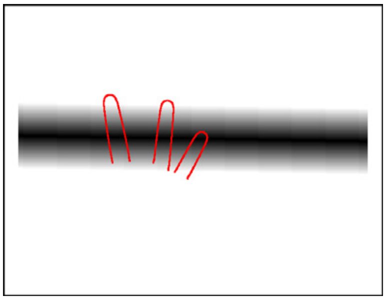


Fig. 18. Fuzzy spatial representation of the alignment constraint, with respect to the *middle finger*. Positions of *index*, *ring* and *little* fingers have already been estimated, and are used to compute a regression line, from which a distance function is defined to constrain the position of *middle finger*.

$$\begin{aligned}
{}^{B}\nu^{angle}(\mathbf{x}_t^m;\tilde{\mathbf{x}}_t) &= \mathop{\psi}_{l=1}^{L} {}^{B}\mu^{angle}_{\tilde{\mathbf{x}}_t^l}(\mathbf{x}_t^m)\\
&= \min\left({}^{B}\mu^{-\pi/8}_{\tilde{\mathbf{x}}_t^{li}}(\mathbf{x}_t^m), {}^{B}\mu^{-\pi/8}_{\tilde{\mathbf{x}}_t^{ri}}(\mathbf{x}_t^m), {}^{B}\mu^{\pi/8}_{\tilde{\mathbf{x}}_t^{in}}(\mathbf{x}_t^m)\right) {}^{Q}\nu^{alignment}_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^m)\\
&= \mathop{\psi}_{l_1=1}^{L} \mathop{\psi}_{l_2=l_1+1}^{L} \mathop{\psi}_{l_3=l_2+1}^{L} {}^{Q}\mu^{alignment}_{\tilde{\mathbf{x}}_t^{l_1},\tilde{\mathbf{x}}_t^{l_2},\tilde{\mathbf{x}}_t^{l_3}}(\mathbf{x}_t^*)\\
&= {}^{Q}\mu^{alignment}_{\tilde{\mathbf{x}}_t^{li},\tilde{\mathbf{x}}_t^{ri},\tilde{\mathbf{x}}_t^{in}}(\mathbf{x}_t^m) {}^{B}\nu^{exclusion}_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^m) = \mathop{\psi}_{l=1}^{L} {}^{B}\mu^{exclusion}_{\tilde{\mathbf{x}}_t^l}(\mathbf{x}_t^m)\\
&= \min\left({}^{B}\mu^{exclusion}_{\tilde{\mathbf{x}}_t^{li}}(\mathbf{x}_t^m), {}^{B}\mu^{exclusion}_{\tilde{\mathbf{x}}_t^{ri}}(\mathbf{x}_t^m), {}^{B}\mu^{exclusion}_{\tilde{\mathbf{x}}_t^{in}}(\mathbf{x}_t^m)\right)
\end{aligned}$$

Note that it is possible to optimize the computations by taking into account dependencies between constraints. In the previous example, the computation of the exclusion between the *middle finger* and the *index finger* and of the orientation are not useful,

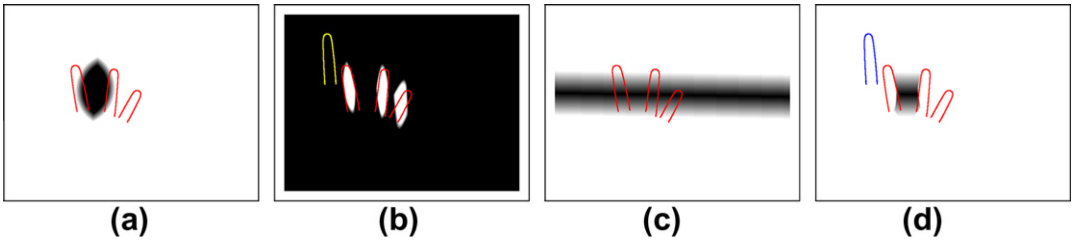
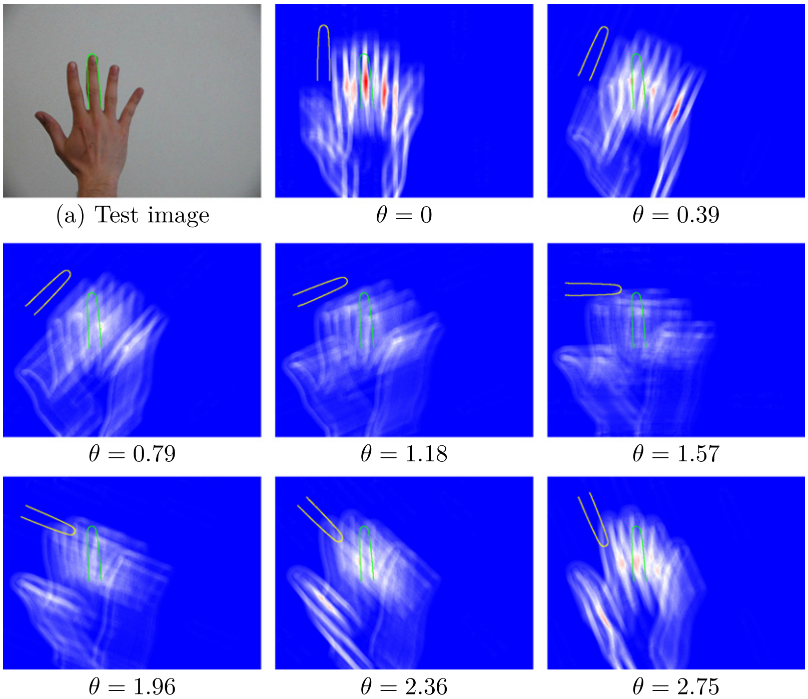


Fig. 19. Spatial constraints on (a) distance, (b) exclusion and (c) alignment, by considering the finger orientation (its representation if superimposed in yellow in the constraint image, and in blue in the final result image). (d) Fusion result for the position of *middle finger*, which shows that only a restricted region satisfying all constraints is obtained.


(a) Test image

$\theta = 0$

$\theta = 0.39$

$\theta = 0.79$

$\theta = 1.18$

$\theta = 1.57$

$\theta = 1.96$

$\theta = 2.36$

$\theta = 2.75$

Fig. 20. Tests on the log-likelihood used for Hand sequence. (a) Test image, in which we want to track the *middle finger* in green. Results using the shape represented in yellow with an angle (in radian, the right value being −0.07) of θ. Red regions correspond to high likelihood values.

because the *ring finger* (processed before) already satisfied the constraints imposed by the *index finger*.

Let us illustrate the spatial constraints in the image domain, on a representative frame of the sequence (see Fig. 14). The spatial constraint on distance is illustrated in Fig. 15. As before, a dark value indicates that the constraint is satisfied. The goal of the distance relation is to constrain the central position of the *middle finger*. Using the formulated constraints for the *index finger*

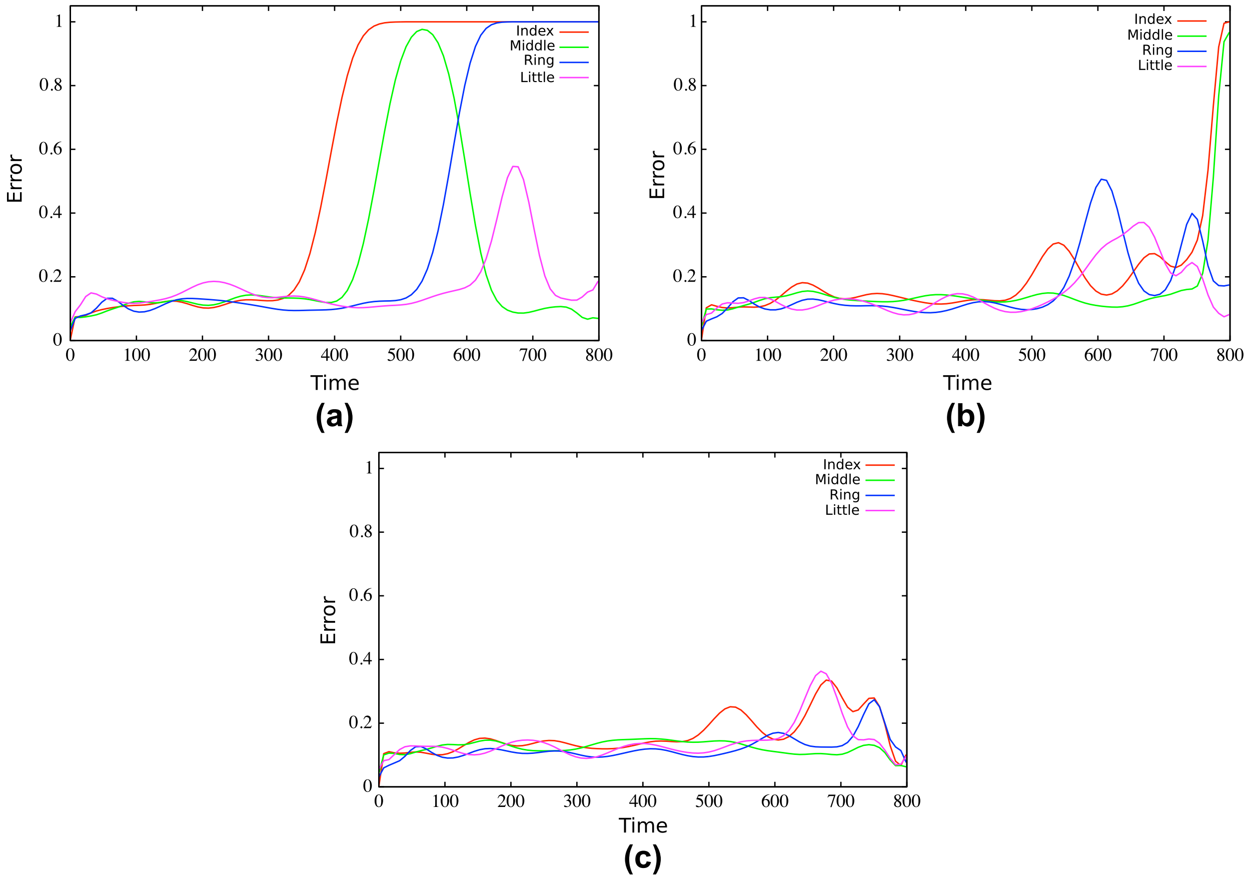


Fig. 21. Shape errors for hand tracking. (a) RPS without spatial constraints. (b) PS with spatial constraints. (c) RPS with spatial constraints.

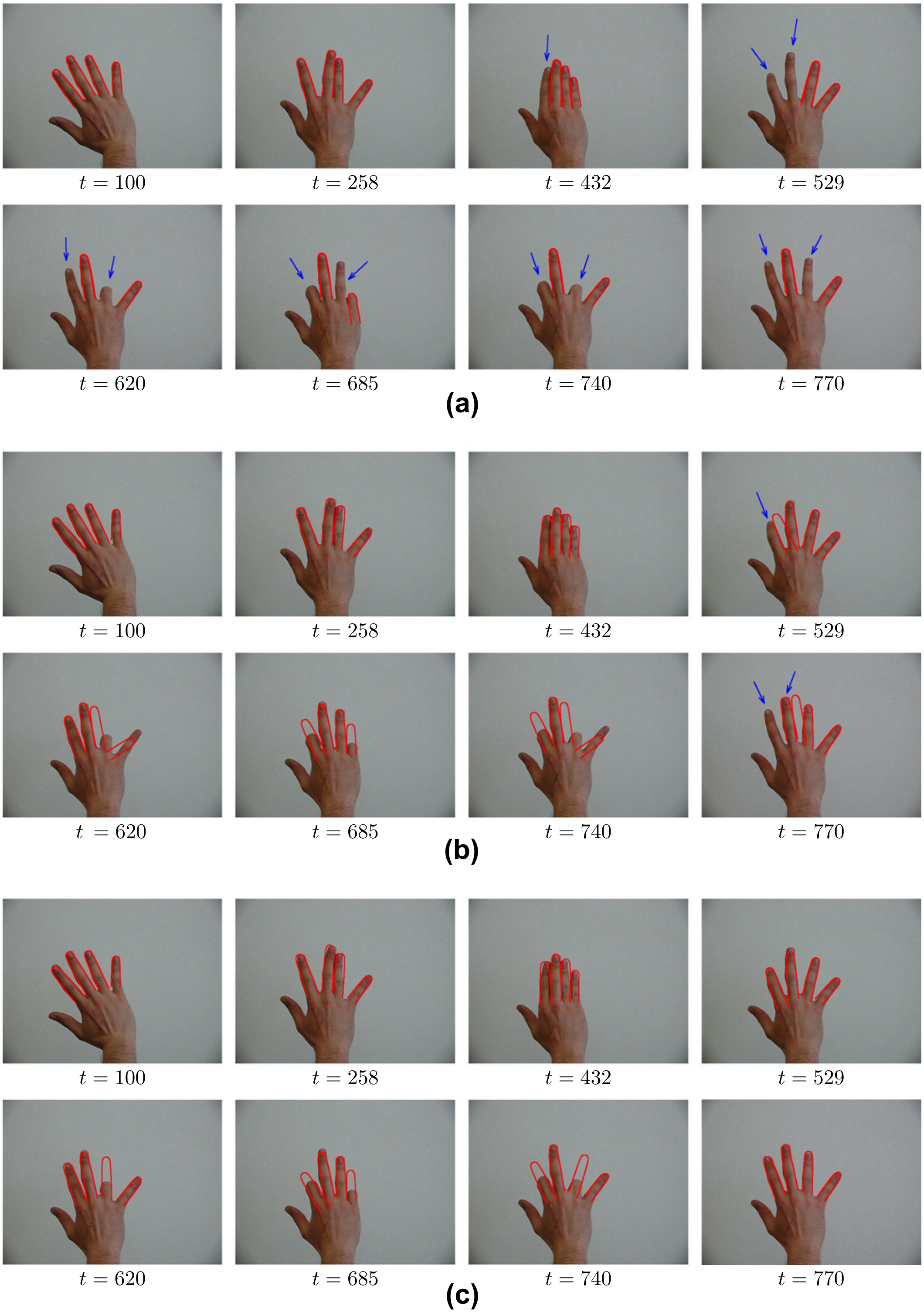


Fig. 22. Hand tracking results at different instants (tracking errors are indicated by blue arrows). (a) RPS without spatial constraints. (b) PS with spatial constraints. (c) RPS with spatial constraints.

(Fig. 15a), the *ring finger* (Fig. 15b), and the *little finger* (Fig. 15c), the fusion is represented in the image domain in Fig. 15d. Note that this spatial representation just aims at illustrating the constraints, and do not need to be actually computed (except for the exclusion principle). The constraints are not evaluated on the whole set of pixels, but just on a specific realization of the *middle finger* state $\mathbf{x}_t^m$.

The spatial constraint on relative orientation is illustrated in Fig. 16. This constraint restricts the possible orientations of the finger. The fusion illustrated in Fig. 16d is actually computed in the angle domain, and not in the spatial one.

The spatial constraint of exclusion is illustrated in Fig. 17. For this illustration, it is necessary to fix the orientation, because the constraint depends on both the orientation and the central position of the finger. The considered orientation is then represented by an hypothetic state (represented in yellow[1]) superimposed on the images of constraints.

The spatial constraint of alignment is illustrated in Fig. 18. The result is directly obtained, without any fusion process, because it is a quaternary operator. A ternary operator would have been used if the positions of just two fingers would be known. When less than two fingers have already been processed, this constraint is not considered in the decision process.

The resulting fusion of the distance, exclusion and alignment constraints, using a fixed orientation, is shown in Fig. 19.

It would have been possible to define other more complex or more robust spatial constraints, such as a finer alignment constraint [46], and, instead of using an exclusion concept, we could have used a model of the concept of "cross" [47]. However, the constraints defined in this experiment permit to represent a lot of possible impossible configurations, while remaining simple to compute. They proved to be sufficient to illustrate the interest and the power of the proposed approach.

The likelihood is derived from the maximum gradient magnitude along the normals at the control points of the B-spline modeling a finger (this is a simplified version of the one proposed in [48]):

$$p(\mathbf{y}_t^i|\mathbf{x}_t^i) \propto \exp\left(\frac{\lambda}{P_{a^i}} \sum_{p=1}^{P_{a^i}} \max_{x \in \overrightarrow{n_{a_p^i}}} \nabla\left[I_t(x) * \mathcal{N}(x, \sigma^2)\right]\right) \tag{26}$$

where ∇ and $*$ are respectively gradient and convolution operators, a_p^i the pth control point of the B-spline of finger i, of length P_{a^i} and defined by $\mathbf{x}_t^i$, $\overrightarrow{n_{a_p^i}}$ the normal at point a_p^i, $I_t(x)$ the gray level value at point x of image I_t at instant t, $\lambda = 30$ a constant empirically fixed, and $\mathcal{N}(x, \sigma^2)$ a Gaussian distribution of mean x and fixed variance $\sigma^2 = 0.1$ to make the observation robust to noise. Fig. 20 shows the corresponding likelihoods. As we want to estimate the position and orientation of the fingers, it is necessary for the likelihood to be robust to both parameters. High values of the likelihood (in red) are observed also for the other fingers, and therefore the tracking has to be constrained, to avoid the tracked finger to be attracted towards other local maxima.

The shape errors obtained for *index*, *middle*, *ring* and *little* fingers with RPS without spatial constraints, PS with spatial constraints, and RPS with spatial constraints are given in Fig. 21, with $N = 2000$ particles. An overlapping ratio is computed as the number of common points between the estimated shape and the ground truth, divided by the maximum of their surfaces. Then shape errors are computed as 1 minus this overlapping ratio. Results using a simple PS strategy were generated using a random sequence order (in this particular sequence, fixed sequence order did not provide quite as good results). RPS with spatial constraints (Fig. 21c) leads to better results than the other methods (Fig. 21a and b), giving less than 36% of shape overlapping errors for all the fingers. The results for the corresponding shapes are displayed in Fig. 22 and have been obtained with $N = 2000$ particles. RPS without spatial constraints fails as soon as a finger is partially hidden, or close to another one. In fact, such a situation generates an ambiguity, because there are several modes in the likelihood (see for example Fig. 22a at time 432). However, RPS provides better results than those obtained with PS thanks to the estimation of the scenario, which allows estimating first fingers that are trusted, i.e. the visible ones, and then the other ones, which are then more constrained by the fuzzy spatial relations (see from $t = 529$ in Figs. 22b and c, and 21b and c). This is particularly visible at the end of the sequence ($t \geqslant 770$), where the *index finger* is wrongly estimated by PS, and is localized over the *middle finger*. Using fuzzy spatial constraints conditions the estimation of the position of the *middle finger*, that is "blocked" between the real *middle* and *ring* fingers (see the last image of Figs. 22b and 21b), which is consistent with the configuration of a hand. Results obtained with PS and RPS with spatial constraints are variable, because of the stochastic nature of the algorithm, and the difficulty of simulation of the sequence (four objects, high constraints on their spatial conditions). However we observed a systematic superiority of RPS over PS.

8. Conclusion

The method proposed in this paper includes two contributions. First, we introduce fuzzy spatial constraints into a multiple object tracking based on particle filtering. This novel information allows us to easily handle constraints between objects, such that *on the right of*, *behind*, *aligned with*, etc. in a unified framework. By modeling unary, binary, or n-ary fuzzy operators, we can evaluate the possibility of a whole configuration of several objects. This is modeled by a fuzzy set that evaluates the degree of satisfaction of a set of constraints, and is then embedded into a probabilistic framework to estimate the probability of an object configuration at a time t. As a second contribution, the multiple object particle filter uses a Ranked Partitioned Sampling strategy, which, as the Partitioned Sampling [10], tackles the problem of dimensionality by sequentially performing a weighted resampling step in single object state spaces. Moreover, the simulation order involved in the proposed RPS is adaptive, which makes the tracking more robust and alleviates the impoverishment effect. Since the Ranked Partitioned Sampling just shifts the simulation order of the objects, it performs the tracking with a computation time identical to the PS one. Concerning the fuzzy relations, it depends on the nature of the spatial constraint. For example, computing an orientation constraint between objects is straightforward since this is just a difference of angles, whereas computing an exclusion constraint requires to label the objects in an image. This can still be done efficiently and a carefully designed implementation should not suffer from computation time drawbacks.

A direct consequence of adding constraints during the likelihood computation rather than during the simulation (which is a common choice in the literature) is that the likelihood can be highly picked, which increases the variance of the particle weights. A simple solution should be to slightly relax these constraints to avoid a possible divergence of the filter. A most sophisticated one, that is not always possible, is to generate the particles conditionally to the constraints. This solution is provided by our model (Eq. (18)), but not considered in our tests because constraints were too complex.

The fuzzy spatial constraint modeling could deal with more complex information, for example those often used in scene description [17,49–51]. We could also automatically learn the parameters of these constraints [52]. A last improvement would be to consider the number M of objects as a random variable [3,53].

Appendix A. Complement about Eq. (16)

Proposition 1. *Eq. (16) defines a probability distribution.*

[1] For interpretation of color in Figs. 1, 3–22, the reader is referred to the web version of this article.

Proof. Let us prove that the sum over all possibilities is equal to 1:

$$\begin{aligned}
&\sum_{h=1}^{M}\mathbb{P}\left(\mathbf{o}_t^{\mathbf{s}^k}=h|\mathbf{o}_{t-1}^{\mathbf{s}^k}\triangleq k,\{\mathbf{o}_t^{\mathbf{s}^u}\}_{u=1}^{k-1}\right)=\sum_{h=1}^{M}\left[1-\sum_{j=1}^{k-1}\delta^h_{\mathbf{o}_t^{\mathbf{s}^j}}\right]\left[\alpha_{k,h}+\frac{1}{M-k+1}\sum_{j=1}^{k-1}\alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}\right]\\
&=\sum_{h=1}^{M}\left[\alpha_{k,h}+\frac{1}{M-k+1}\sum_{j=1}^{k-1}\alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}-\alpha_{k,h}\sum_{j=1}^{k-1}\delta^h_{\mathbf{o}_t^{\mathbf{s}^j}}-\frac{1}{M-k+1}\left[\sum_{j=1}^{k-1}\alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}\right]\left[\sum_{j=1}^{k-1}\delta^h_{\mathbf{o}_t^{\mathbf{s}^j}}\right]\right]\\
&=\sum_{h=1}^{M}[\alpha_{k,h}]+\frac{1}{M-k+1}\sum_{h=1}^{M}\left[\sum_{j=1}^{k-1}\alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}\right]-\sum_{h=1}^{M}\left[\sum_{j=1}^{k-1}\alpha_{k,h}\delta^h_{\mathbf{o}_t^{\mathbf{s}^j}}\right]-\frac{1}{M-k+1}\sum_{h=1}^{M}\left[\left[\sum_{j=1}^{k-1}\alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}\right]\left[\sum_{j=1}^{k-1}\delta^h_{\mathbf{o}_t^{\mathbf{s}^j}}\right]\right]\\
&=1+\frac{M}{M-k+1}\sum_{j=1}^{k-1}\alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}-\sum_{j=1}^{k-1}\left[\sum_{h=1}^{M}\alpha_{k,h}\delta^h_{\mathbf{o}_t^{\mathbf{s}^j}}\right]-\frac{1}{M-k+1}\left[\sum_{j=1}^{k-1}\alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}\right]\left[\sum_{j=1}^{k-1}\sum_{h=1}^{M}\delta^h_{\mathbf{o}_t^{\mathbf{s}^j}}\right]\\
&=1+\frac{M}{M-k+1}\sum_{j=1}^{k-1}\alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}-\sum_{j=1}^{k-1}\alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}-\frac{k-1}{M-k+1}\left[\sum_{j=1}^{k-1}\alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}\right]\\
&=1+\sum_{j=1}^{k-1}\alpha_{k,\mathbf{o}_t^{\mathbf{s}^j}}\underbrace{\left[\frac{M}{M-k+1}-1-\frac{k-1}{M-k+1}\right]}_{=0}=1\square
\end{aligned}$$

References

[1] A. Doucet, B. Vo, C. Andrieu, M. Davy, Particle filtering for multi-target tracking and sensor management, in: Fifth International Conference on Information Fusion, vol. 1, 2002, pp. 474–481.

[2] D. Schulz, W. Burgard, D. Fox, A. Cremers, Tracking multiple moving targets with a mobile robot using particle filters and statistical data association, in: IEEE International Conference on Robotics and Automation, vol. 2, 2001, pp. 1665–1670.

[3] C. Hue, J.-P. Le Cadre, P. Pérez, Sequential Monte Carlo methods for multiple target tracking and data fusion, IEEE Transactions on Signal Processing 50 (2) (2002) 309–325.

[4] C. Kreucher, A. Hero, K. Kastella, Multiple model particle filtering for multi-target tracking, in: The 12th Annual Workshop on Adaptive Sensor Array Processing, 2004.

[5] C. Kreucher, K. Kastella, A. Hero, Multitarget tracking using the joint multitarget probability density, IEEE Transactions on Aerospace and Electronic Systems 41 (4) (2005) 1396–1414.

[6] K. Okuma, A. Taleghani, N. de Freitas, J. Little, D. Lowe, A boosted particle filter: multitarget detection and tracking, in: European Conference on Computer Vision, vol. 3021, 2004, pp. 28–39.

[7] J. Vermaak, S. Godsill, P. Pérez, Monte Carlo filtering for multi-target tracking and data association, IEEE Transactions on Aerospace and Electronic Systems 41 (2005) 309–332.

[8] D.J.C. MacKay, Introduction to Monte Carlo methods, in: NATO Advanced Study Institute on Learning in Graphical Models, 1998, pp. 175–204.

[9] C.P. Robert, G. Casella, Monte Carlo Statistical Methods, Springer-Verlag, 2005.

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

[11] J. MacCormick, M. Isard, Partitioned sampling, articulated objects, and interface-quality hand tracking, in: European Conference on Computer Vision-Part II, 2000, pp. 3–19.

[12] K. Smith, D. Gatica-Perez, Order matters: a distributed sampling method for multi-object tracking, in: British Maschine Vision Conference, 2004, pp. 25–32.

[13] S. Duffner, J. Odobez, E. Ricci, Dynamic partitioned sampling for tracking with discriminative features, in: British Maschine Vision Conference, 2009.

[14] Z. Khan, T. Balch, F. Dellaert, MCMC-based particle filtering for tracking a variable number of interacting targets, IEEE Transactions Pattern Analysis and Machine Intelligence 27 (11) (2005) 1805–1918.

[15] J. Freeman, The modelling of spatial relations, Computer Graphics and Image Processing 4 (2) (1975) 156–171.

[16] I. Bloch, Fuzzy spatial relationships for image processing and interpretation: a review, Image and Vision Computing 23 (2) (2005) 89–110.

[17] O. Colliot, O. Camara, I. Bloch, Integration of fuzzy spatial relations in deformable models – application to brain MRI segmentation, Pattern Recognition 39 (8) (2006) 1401–1414.

[18] G. Fouquier, J. Atif, I. Bloch, Local reasoning in fuzzy attribute graphs for optimizing sequential segmentation, in: Graph Based Representations in Pattern Recognition, vol. LNCS 4538, 2007, pp. 138–147.

[19] C. Hudelot, J. Atif, I. Bloch, Fuzzy spatial relation ontology for image interpretation, Fuzzy Sets and Systems 159 (15) (2008) 1929–1951.

[20] D. Pham, Fuzzy clustering with spatial constraints, in: IEEE International Conference on Image Processing, 2002, pp. 65–68.

[21] N. Widynski, S. Dubuisson, I. Bloch, Integration of fuzzy spatial information in tracking based on particle filtering, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 41 (3) (2011) 635–649.

[22] N. Widynski, S. Dubuisson, I. Bloch, Introducing fuzzy spatial constraints in a ranked partitioned sampling for multi-object tracking, in: International Symposium on Visual Computing, 2010, pp. 393–404.

[23] A. Doucet, N. De Freitas, N. Gordon (Eds.), Sequential Monte Carlo Methods in Practice, Springer, 2001.

[24] D. Avitzour, Stochastic simulation Bayesian approach to multitarget tracking, IEEE Proceedings on Radar, Sonar and Navigation 142 (2) (1995) 41–44.

[25] M. Yang, T. Yu, Y. Wu, Game-theoretic multiple target tracking, in: International Conference on Computer Vision, 2007.

[26] M. Isard, J. MacCormick, BraMBLe: A Bayesian multiple-blob tracker, in: International Conference on Computer Vision, vol. 2, 2001, pp. 34–41.

[27] P. Pérez, J. Vermaak, Bayesian tracking with auxiliary discrete processes. application to detection and tracking of objects with occlusions, in: International Conference on Computer Vision – Workshop on Dynamical Vision, 2005, pp. 190–202.

[28] C. Snyder, T. Bengtsson, P. Bickel, J. Anderson, Obstacles to high-dimensional particle filtering, Monthly Weather Review 136 (12) (2008) 4629–4640.

[29] M. Bray, E. Koller-Meier, L.V. Gool, Smart particle filtering for high-dimensional tracking, Computer Vision and Image Understanding 106 (1) (2007) 116–129.

[30] N. Vaswani, Particle filtering for large-dimensional state spaces with multimodal observation likelihoods, IEEE Transactions on Signal Processing 56 (10) (2008) 4583–4597.

[31] A. Doucet, N. De Freitas, K. Murphy, S. Russell, Rao-Blackwellised particle filtering for dynamic Bayesian networks, in: Conference on Uncertainty in Artificial Intelligence, 2000, pp. 176–183.

[32] J. MacCormick, Probabilistic modelling and stochastic algorithms for visual localisation and tracking, Ph.D. thesis, University of Oxford, 1999.

[33] N.J. Gordon, D.J. Salmond, A.F.M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, IEEE Proceedings on Radar and Signal Processing 140 (2) (1993) 107–113.

[34] J.S. Liu, R. Chen, Blind deconvolution via sequential imputations, Journal of the American Statistical Association 90 (430) (1995) 567–576.

[35] K.C. Smith, Bayesian methods for visual multi-object tracking with applications to human activity recognition, Ph.D. thesis, École Polytechnique Fédérale de Lausanne, 2007.

[36] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning - I, Information Sciences 8 (3) (1975) 199–249.

[37] I. Bloch, Information combination operators for data fusion: a comparative review with classification, IEEE Transactions on Systems, Man, and Cybernetics, Part A 26 (1) (1996) 52–67.

[38] D. Dubois, H. Prade, Fuzzy Sets and Systems: Theory and Applications, Academic Press, Inc., 1980.

[39] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems 1 (1) (1978) 3–28.

[40] L.A. Zadeh, Probability measures of fuzzy events, Journal of Mathematical Analysis and Applications 23 (2) (1968) 421–427.

[41] J. MacCormick, A. Blake, Spatial dependence in the observation of visual contours, in: European Conference on Computer Vision, vol. 2, 1998, pp. 765–781.

[42] C. Hue, Méthodes séquentielles de Monte-Carlo pour le filtrage non linéaire multi-objets dans un environnement bruité. Applications au pistage multi-cibles et à la trajectographie d'entités dans des séquences d'images 2d, Ph.D. thesis, Université de Rennes I, mention informatique, 2003.

[43] F. Fleuret, J. Berclaz, R. Lengagne, P. Fua, Multi-camera people tracking with a probabilistic occupancy map, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2) (2008) 267–282.

[44] P. Pérez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, in: European Conference on Computer Vision-Part I, 2002, pp. 661–675.

[45] A. El Abed, S. Dubuisson, D. Bereziat, Enmim: energetic normalized mutual information model for online multiple object tracking with unlearned motions, in: International Conference on Advanced Concepts for Intelligent Vision Systems, 2007, pp. 955–967.

[46] M.-C. Vanegas, I. Bloch, J. Inglada, Searching aligned groups of objects with fuzzy criteria, in: International Conference on Information Processing and management of Uncertainty in Knowledge-Based Systems, IPMU, 2010, pp. 605–613.

[47] M. Vanegas, I. Bloch, J. Inglada, Fuzzy spatial relations for high resolution remote sensing image analysis: the case of "To Go Across", in: IEEE International Geoscience and Remote Sensing Symposium, vol. 4, 2009, pp. 773–776.

[48] M. Isard, A. Blake, Condensation – conditional density propagation for visual tracking, International Journal of Computer Vision 29 (1) (1998) 5–28.

[49] O. Nempont, J. Atif, E. Angelini, I. Bloch, Structure segmentation and recognition in images guided by structural constraint propagation, in: European Conference on Artificial Intelligence, 2008, pp. 621–625.

[50] A. Deruyver, Y. Hodé, Constraint satisfaction problem with bilevel constraint: application to interpretation of over-segmented images, Artificial Intelligence 93 (1–2) (1997) 321–335.

[51] A. Deruyver, Y. Hodé, E. Leammer, J. Jolion, Adaptive pyramid and semantic graph: knowledge driven segmentation, in: Graph-based representations in pattern recognition, 2005, pp. 213–222.

[52] J. Atif, C. Hudelot, G. Fouquier, I. Bloch, E. Angelini, From generic knowledge to specific reasoning for medical image interpretation using graph-based representations, in: International Joint Conference on Artificial Intelligence, 2007, pp. 224–229.

[53] K. Smith, D. Gatica-Perez, J.-M. Odobez, Using particles to track varying numbers of objects, in: Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 962–969.