Multiple-Target Tracking based on Particle Filtering*

Daniel Rowe
Computer Vision Centre

February 13, 2008

Here, tracking is performed by enhancing the particle filtering framework. This approach has been widely explored by several previous algorithms, as discussed before. Despite this effort, many undesirable effects still remain. These are here highlighted, and some proposals are presented in order to cope with them.

1 Framework Outline

A probabilistic framework is commonly used as a way to perform tracking in order to deal with uncertainty over time [23]. Classical approaches, such as the *Kalman Filter* [14], rely on linearity and Gaussianity assumptions about the involved distributions.

More recent works make use of *Bayesian filters* combined with *Monte Carlo Simulation* methods in order to deal with nonlinear and non-Gaussian transition models and non-Gaussian likelihood functions [22, 18]. Subsequent developments have introduced a re-sampling phase in the sequential simulation-based Bayesian filter algorithms [8]. These approaches are known as *particle filtering* within the control field or *survival of the fittest* in Artificial Intelligence.

Such methods were first introduced in the computer-vision research area by Isard and Blake, and renamed as *Condensation* [10, 11]. They have been widely used in recent years [12, 3, 26, 16, 24, 17, 13, 20, 19, 27, 4]. Excellent reviews have been presented by Doucet [5], and by Arulampalam et al. [1]. Further, comprehensive treatments are given in [6, 21]. However, several important drawbacks remain, as stated by King and Forsyth [15]. Despite the great number of improvements that have been already introduced, many open issues prevent from stating that particle filters are able to solve unconstrained tracking problems.

*From *Towards Robust Multiple-Target Tracking in Unconstrained Human-Populated Environments.* Daniel Rowe. PhD Thesis, Chapter 4, Universitat Autònoma de Barcelona, Spain, 2008.

2 Probabilistic Framework

From a probabilistic point of view, the tracking problem involves dealing with stochastic processes. These are series of time-slices describing the state of all entities within the scene. Each time-slice consists of a set of random variables[1]. Two kind of variables can be distinguished, namely unobservable state variables at time t, denoted as $\mathbf{S}_t$, and observable evidence variables, denoted as $\mathbf{E}_t$. The interval between time-slices depends on the frame rate[2].

In order to specify the dependencies among the different variables, these are ordered following a temporal criterion, i.e, taking causality into account. This means that the variables from previous time-slices cause the values of subsequent time-slice variables. Thus, it should be possible to specify conditional probability density functions for all variables given their predecessors, from now on called *parents* [23]. On the order hand, variable conditional independence within a time-slice could be established given a set of parents.

However, since every time-slice must be considered, several problems arise:

1. There is an unbounded set of conditional probability density functions.

   This problem can be overcome making the *homogeneous process assumption*:

   The process is governed by laws that do not change themselves over time.

   Hence, there is no need to specify all conditional pdf but only those within a representative time-slice.

2. There is an unbounded set of parents.

   Let us consider separately the effect of the parents on the state variables $\mathbf{S}_t$ and on evidence variables $\mathbf{E}_t$. Considering the *Markov assumption* on both states and evidences, it is possible to get over this problem:

   (a) *The current state* $\mathbf{S}_t$ *depends only on a finite history of previous states,* $\mathbf{S}_{t-\tau:t-1}$.

   Therefore, the *state* could be defined as the information needed to make the future independent from the past given the present. In *first-order Markov processes* the current state only depends on the immediately previous one. Here, this kind of Markov processes is

[1] The following notation is here used: related to variables, non-bold lowercase denotes scalars, whereas bold lowercase denotes vectors, and matrices are given by bold uppercase. In a probabilistic context, uppercase denotes probability density functions (pdf) and random variables; lowercase denotes probabilities and variable instances. $\mathbf{X}_{t_1:t_2}$ denotes a variable set from time $t = t_1$ to $t = t_2$.

[2] This parameter is set considering the possible dynamics of the targets that could appear in the scene.

considered, since it is always possible to reformulate a non first-order Markov process as a first-order one by increasing the state variable set [23].
Thus, the state variables are conditional independent of all other previous variables given the previous state:

$$P\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{E}_{1:t-1}\right) = P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right). \tag{1}$$

The latter conditional pdf is called the *transition model*. In the tracking problem here presented, the transition model will be split into a *dynamic model*, which considers the target's motion, and an *aspect model*, which captures the target's shape and appearance.

(b) *The evidence variables at time t* $\mathbf{E}_t$ *depend only on the current state* $\mathbf{S}_t$.

Hence, the evidence variables are conditional independent from all other variables given the state:

$$P\left(\mathbf{E}_t \mid \mathbf{S}_{0:t-1}, \mathbf{E}_{1:t-1}\right) = P\left(\mathbf{E}_t \mid \mathbf{S}_t\right). \tag{2}$$

In this case, the latter conditional pdf is called the *observation* or *sensor model*. It is also called the *likelihood* function since it forecasts how likely an observation is, once the state is given. It models a causal relation: it is the current state which causes the obtained evidence.

Thus, the developments within the scene can be modelled as a Hidden Markov Model (HMM) where $\mathbf{S}_t$ constitutes the unobservable or hidden state variables and $\mathbf{E}_t$ the observable evidence variables at time t. The HMM is described by:

- an initial prior state density function, $P\left(\mathbf{S}_0\right)$;
- the transition model[3], $P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right)$ for $t \geq 1$;
- the likelihood function, $P\left(\mathbf{E}_t \mid \mathbf{S}_t\right)$ for $t \geq 1$;
- both assumptions on variable conditional independence stated in Eqs. (1) and (2):
  - the state variables, $\{\mathbf{S}_t; t \in \mathbb{N}\}, \mathbf{S}_t \in \mathbb{R}^{n_s}$, given the immediately previous state $\mathbf{S}_{t-1}$; $n_\mathbf{s}$ denotes the state-space dimension;
  - the evidence variables, $\{\mathbf{E}_t; t \in \mathbb{N}\}, \mathbf{E}_t \in \mathbb{R}^{n_e}$, given the corresponding state variable; $n_\mathbf{e}$ denotes the evidence-space dimension.

[3] A sequence of random variables $\mathbf{S}_t$ satisfying the Markov assumption is called a *Markov chain*. If the conditional probability density functions $P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right)$ are time independent, the Markov chain is called *homogeneous*. However, it does not mean that the probability density functions of consecutive states are the same, $P\left(\mathbf{S}_t\right) = P\left(\mathbf{S}_{t-1}\right)$, a fact that is called *stationarity*.

Given both models and assumptions, it is possible to specify the complete joint density function:

$$\begin{aligned}
P(\mathbf{S}_{0:t}, \mathbf{E}_{1:t}) &= P(\mathbf{E}_t \mid \mathbf{S}_{0:t}, \mathbf{E}_{1:t-1})\, P(\mathbf{S}_{0:t}, \mathbf{E}_{1:t-1}) && \text{(cond. prob.)} \\
&= P(\mathbf{E}_t \mid \mathbf{S}_t)\, P(\mathbf{S}_{0:t}, \mathbf{E}_{1:t-1}) && \text{(Markov on ev.)} \\
&= P(\mathbf{E}_t \mid \mathbf{S}_t)\, P(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{E}_{1:t-1})\, P(\mathbf{S}_{0:t-1}, \mathbf{E}_{1:t-1}) && \text{(cond. prob.)} \\
&= P(\mathbf{E}_t \mid \mathbf{S}_t)\, P(\mathbf{S}_t \mid \mathbf{S}_{t-1})\, P(\mathbf{S}_{0:t-1}, \mathbf{E}_{1:t-1}) && \text{(Markov)} \\
&\dots \\
&= P(\mathbf{S}_0) \prod_{k=1}^{t} P(\mathbf{E}_k \mid \mathbf{S}_k)\, P(\mathbf{S}_k \mid \mathbf{S}_{k-1}), && (3)
\end{aligned}$$

which specifies the probability of every event within the scene and, therefore, can answer every probabilistic query about it. Unfortunately, it is usually too complex to be analytically computed.

3 Bayesian Filtering

Let us now consider the probabilistic inference problem in which the state variable set $\mathbf{S}_{1:t}$ is estimated from the observed evidence $\mathbf{e}_{1:\tau}$, finding out the *posterior probability density function* $P(\mathbf{S}_{1:t} \mid \mathbf{e}_{1:\tau})$. Let us also focus in one of the posterior pdf marginals, $P(\mathbf{S}_t \mid \mathbf{e}_{1:\tau})$.

The previous computation is called *smoothing* if $t < \tau$, *filtering* or *monitoring* if $t = \tau$, and *predicting* if $t > \tau$. The general term *estimating* comprises all three processes. This work is focused on filtering, the computation of the belief state $\mathbf{S}_t$ —or, even better, the posterior pdf over the current state $P(\mathbf{S}_t \mid \mathbf{e}_{1:t})$— given all evidence up to date $\mathbf{e}_{1:t}$.

In this case, instead of the causal relation given by the likelihood function which assigns probabilities to potential evidences given the state, the filtered pdf allows to make and inference about the state given the evidence.

This pdf can be calculated through *recursive estimation*, that is, computing the new posterior given the previous one and the new evidence [5, 23]:

$$\begin{aligned}
P(\mathbf{S}_t \mid \mathbf{e}_{1:t}) &= P(\mathbf{S}_t \mid \mathbf{e}_{1:t-1}, \mathbf{e}_t) && (4) \\
&\propto P(\mathbf{e}_t \mid \mathbf{S}_t, \mathbf{e}_{1:t-1})\, P(\mathbf{S}_t \mid \mathbf{e}_{1:t-1}) && \text{(Bayes')} \\
&= P(\mathbf{e}_t \mid \mathbf{S}_t)\, P(\mathbf{S}_t \mid \mathbf{e}_{1:t-1}) && \text{(Mark. on ev.)} \\
&= P(\mathbf{e}_t \mid \mathbf{S}_t) \int P(\mathbf{S}_t \mid \mathbf{s}_{t-1}, \mathbf{e}_{1:t-1})\, P(\mathbf{s}_{t-1} \mid \mathbf{e}_{1:t-1})\, d\mathbf{s}_{t-1} && \text{(cond.)} \\
&= \underbrace{\underbrace{P(\mathbf{e}_t \mid \mathbf{S}_t)}_{\text{likelihood}}}_{\text{updating}} \underbrace{\int \underbrace{P(\mathbf{S}_t \mid \mathbf{s}_{t-1})}_{\text{trans. model}} \underbrace{P(\mathbf{s}_{t-1} \mid \mathbf{e}_{1:t-1})}_{\text{previous post.}} d\mathbf{s}_{t-1}}_{\text{prediction}}. && \text{(Markov)}
\end{aligned}$$

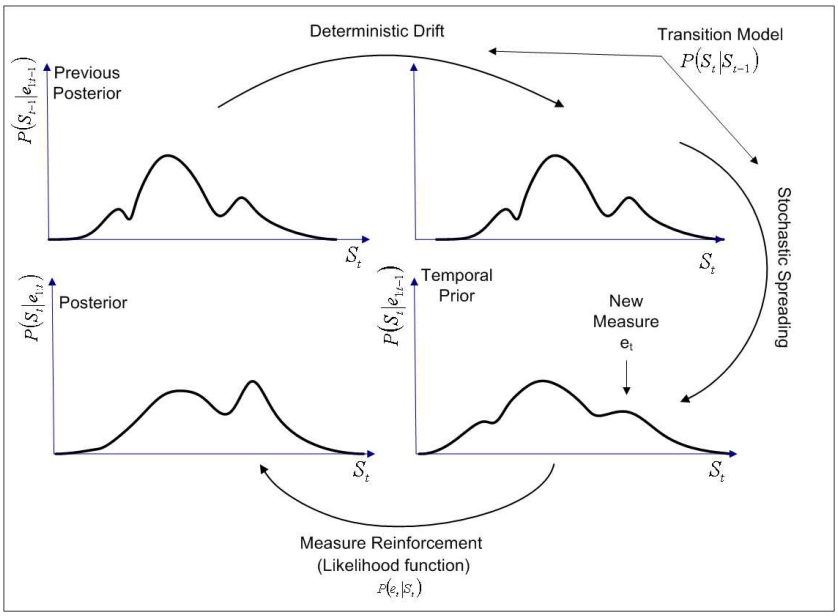

Figure 1: Temporal propagation of posterior density functions. A deterministic drift and a stochastic spreading given by the transition model yield the temporal prior. Then, the new posterior is obtained by using the correction given by likelihood function.

The pdf is projected forward according to the transition model, making a prediction. Then, it is updated in agreement with the new evidence, $\mathbf{e}_t$. The prediction term represents the density function after applying the transition model to the previous posterior density function. It leads to the so-called *prior density function*, $P(\mathbf{S}_t \mid \mathbf{e}_{1:t-1})$. It is called prior because it is previous to the likelihood correction.

The temporal propagation of the posterior pdf marginal can be seen as a diffusion–reinforcement process, see Fig. 1. The transition model has a deterministic and a stochastic component. The former imposes a drift to the probability density function, while the latter causes the spreading of the pdf that increases the state uncertainty. Subsequently, the likelihood function reinforces the pdf in the vicinity of observations altering the peaks and reducing the uncertainty.

4 Monte-Carlo Simulation

Unfortunately, the recursive estimation given above leads to expressions that are impossible to evaluate analytically unless strong assumptions are made. For example, the Kalman Filter is a linear recursive estimator which assumes a

linear Gaussian transition model, and a Gaussian likelihood function.

In a more general framework, this problem is overcome by making use of Monte-Carlo methods[4], where N independent-and-identically-distributed (i.i.d.) random samples, $\left\{\mathbf{s}_t^i; i = 1 : N\right\}$, are generated from the posterior pdf, $P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right)$.

On the one hand, a simulated probability density function is given by the following expression:

$$\tilde{P}\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) = \frac{1}{N}\sum_{i=1}^{N} \delta\left(\mathbf{S}_t - \mathbf{s}_t^i\right), \tag{5}$$

where $\delta\left(\cdot\right)$ denotes the Dirac delta function.

On the other hand, the posterior expectation is given by:

$$\mu \triangleq \mathbb{E}_{P(\mathbf{S}_t|\mathbf{e}_{1:t})}\left[\mathbf{S}_t\right] = \int \mathbf{S}_t P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) d\mathbf{S}_t, \tag{6}$$

and the posterior variance by:

$$\sigma^2 \triangleq \mathbb{E}_{P(\mathbf{S}_t|\mathbf{e}_{1:t})}\left[\mathbf{S}_t^2\right] - \mathbb{E}^2_{P(\mathbf{S}_t|\mathbf{e}_{1:t})}\left[\mathbf{S}_t\right]. \tag{7}$$

Let us now consider the following estimate:

$$\bar{\mathbf{S}}_N = \int \mathbf{S}_t \tilde{P}\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) d\mathbf{S}_t = \frac{1}{N}\sum_{i=1}^{N} \mathbf{s}_t^i, \tag{8}$$

if both posterior expectation and variance are finite, it follows, due to the Central Limit Theorem, that when $N \to \infty$, $\bar{\mathbf{S}}_N$ has a distribution that is approximately normal, which mean is the posterior expectation μ and its variance is proportional to the posterior variance σ^2:

$$\bar{\mathbf{S}}_N - \mu \dot{\sim} \mathcal{N}\left(0, \frac{\sigma^2}{N}\right). \tag{9}$$

Therefore, the posterior expectation $\mathbb{E}_{P(\mathbf{S}_t|\mathbf{e}_{1:t})}\left[\mathbf{S}_t\right]$ can be estimated and, in addition, the deviation from the true value follows a normal distribution. Moreover, the higher the number of samples is, the lower the estimate variance will be. These results are also applied for expectations of the form:

$$\mathbb{E}_{P(\mathbf{S}_t|\mathbf{e}_{1:t})}\left[\phi\left(\mathbf{S}_t\right)\right] = \int \phi\left(\mathbf{S}_t\right) P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) d\mathbf{S}_t \tag{10}$$

where $\phi\left(\cdot\right)$ is a general function of the state.

However, there are several drawbacks which prevent from using the method as it is presented above. The posterior pdf, $P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right)$, is usually complex

[4]Stochastic simulation techniques are referred as Monte-Carlo methods for the Casinos of Monte Carlo, the capital city of gambles. Roulette wheels and dice rolls are simple random number generators.

enough, multivariate, and only known up to a proportionality constant. These problems make impossible to sample directly from it. Thus, alternative solutions are required.

5 Sequential Importance Sampling (SIS)

It is possible to avoid the difficulty of sampling directly from the posterior density by sampling from an importance or proposal distribution, $Q(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})$. As it will be proved, the posterior density function can be approximated arbitrary well by drawing samples from a proposal distribution, and thereby, obtaining approximations of the expectations of interest. Without the lack of generality, results are here obtained for the first raw moment, i.e, the mean:

$$\begin{aligned}
\mu_{P(\mathbf{S}_{0:t}|\mathbf{e}_{1:t})} &= \int \mathbf{S}_{0:t} P(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})\, d\mathbf{s}_{0:t} && (11)\\
&= \int \mathbf{S}_{0:t} \frac{P(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})}{Q(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})} Q(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})\, d\mathbf{s}_{0:t} && \text{(proposal distr.)}\\
&= \int \mathbf{S}_{0:t} \frac{P(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t})\, P(\mathbf{S}_{0:t})}{P(\mathbf{e}_{1:t})\, Q(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})} Q(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})\, d\mathbf{s}_{0:t}. && \text{(Bayes)}
\end{aligned}$$

By defining the *unnormalised importance weights* as:

$$\pi_t = \frac{P(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t})\, P(\mathbf{S}_{0:t})}{Q(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})}, \quad (12)$$

and conditioning over the evidence probability density function, it follows that:

$$\begin{aligned}
\mu_{P(\mathbf{S}_{0:t}|\mathbf{e}_{1:t})} &= \frac{1}{P(\mathbf{e}_{1:t})} \int \mathbf{S}_{0:t} \pi_t Q(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})\, d\mathbf{s}_{0:t} && (13)\\
&= \frac{\int \mathbf{S}_{0:t} \pi_t Q(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})\, d\mathbf{s}_{0:t}}{\int P(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t})\, P(\mathbf{S}_{0:t})\, d\mathbf{s}_{0:t}} && \text{(conditioning)}\\
&= \frac{\int \mathbf{S}_{0:t} \pi_t Q(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})\, d\mathbf{s}_{0:t}}{\int P(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t})\, P(\mathbf{S}_{0:t}) \frac{Q(\mathbf{S}_{0:t}|\mathbf{e}_{1:t})}{Q(\mathbf{S}_{0:t}|\mathbf{e}_{1:t})} d\mathbf{s}_{0:t}} && \text{(prop. distr.)}\\
&= \frac{\int \mathbf{S}_{0:t} \pi_t Q(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})\, d\mathbf{s}_{0:t}}{\int \pi_t Q(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t})\, d\mathbf{s}_{0:t}} && \text{(weight def.)}\\
&= \frac{\mathbb{E}_{Q(\mathbf{S}_t|\mathbf{e}_{1:t})}[\mathbf{S}_{0:t} \pi_t]}{\mathbb{E}_{Q(\mathbf{S}_t|\mathbf{e}_{1:t})}[\pi_t]}. && \text{(expect. def.)}
\end{aligned}$$

Both expectations can be approximated by sampling from the proposal distribution. Thus, the posterior distribution mean is thereby approximated using the following estimate:

$$\begin{aligned}\bar{\mathbf{S}}_N &= \frac{\frac{1}{N}\sum_{i=1}^{N}\mathbf{s}_{0:t}^i\pi_t^i}{\frac{1}{N}\sum_{i=1}^{N}\pi_t^i} \\ &= \sum_{i=1}^{N}\mathbf{s}_{0:t}^i\overline{\pi}_t^i, \end{aligned} \tag{14}$$

where:

$$\overline{\pi}_t^i = \frac{\pi_t^i}{\sum_{j=1}^{N}\pi_t^j}, \tag{15}$$

denotes the *normalised importance weights*. The posterior density function can then be approximated in the following way:

$$\begin{aligned} P\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) &\approx \tilde{P}\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) \\ &\approx \sum_{i=1}^{N}\overline{\pi}_t^i\delta\left(\mathbf{S}_{0:t} - \mathbf{s}_{0:t}^i\right), \end{aligned} \tag{16}$$

what results from comparing Eq. (8) and Eq. (14).

Considering a filtering scenario, that is, assuming that current states will not be modified by future observations, the proposal distribution can be decomposed as:

$$\begin{aligned} Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right) &= Q\left(\mathbf{S}_{0:t-1}, \mathbf{S}_t \mid \mathbf{e}_{1:t}\right) \\ &= Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) Q\left(\mathbf{S}_{0:t-1} \mid \mathbf{e}_{1:t}\right) \quad \text{(cond. prob.)} \\ &= Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) Q\left(\mathbf{S}_{0:t-1} \mid \mathbf{e}_{1:t-1}\right) \quad \text{(Mark. on ev.)} \end{aligned} \tag{17}$$

This allows us to obtain a recursive expression for the importance weights:

$$\begin{aligned} \pi_t &= \frac{P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right)}{Q\left(\mathbf{S}_{0:t} \mid \mathbf{e}_{1:t}\right)} \\ &= \frac{P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right)}{Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) Q\left(\mathbf{S}_{0:t-1} \mid \mathbf{e}_{1:t-1}\right)} \quad \text{(proposal decomp.)} \\ &= \frac{P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right)}{Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) Q\left(\mathbf{S}_{0:t-1} \mid \mathbf{e}_{1:t-1}\right)} \frac{\pi_{t-1}}{\frac{P(\mathbf{e}_{1:t-1}|\mathbf{S}_{1:t-1})P(\mathbf{S}_{0:t-1})}{Q(\mathbf{S}_{0:t-1}|\mathbf{e}_{1:t-1})}} \quad \text{(weight def.)} \end{aligned} \tag{18}$$

$$
\begin{aligned}
&= \pi_{t-1} \frac{P\left(\mathbf{e}_{1:t} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_{0:t}\right)}{Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) P\left(\mathbf{e}_{1:t-1} \mid \mathbf{S}_{1:t-1}\right) P\left(\mathbf{S}_{0:t-1}\right)} \\
&= \pi_{t-1} \frac{P\left(\mathbf{e}_t \mid \mathbf{S}_{1:t}, \mathbf{e}_{1:t-1}\right) P\left(\mathbf{e}_{1:t-1} \mid \mathbf{S}_{1:t}\right) P\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}\right) P\left(\mathbf{S}_{0:t-1}\right)}{Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) P\left(\mathbf{e}_{1:t-1} \mid \mathbf{S}_{1:t-1}\right) P\left(\mathbf{S}_{0:t-1}\right)} \quad \text{(cond. prob)} \\
&= \pi_{t-1} \frac{P\left(\mathbf{e}_t \mid \mathbf{S}_t\right) P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right)}{Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right)} \quad \text{(Markov)},
\end{aligned}
$$

where

- $P\left(\mathbf{e}_t \mid \mathbf{S}_t\right)$ is the likelihood function;
- $P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right)$ is the transition model;
- and, $Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right)$ is the proposal distribution.

A common and easy choice for the proposal distribution —for instance, the one taken in [11]— is:

$$
Q\left(\mathbf{S}_t \mid \mathbf{S}_{0:t-1}, \mathbf{e}_{1:t}\right) \approx P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right). \tag{19}
$$

In this case, the importance weights are given by:

$$
\pi_t = \pi_{t-1} P\left(\mathbf{e}_t \mid \mathbf{S}_t\right), \tag{20}
$$

and the normalised importance weights are given by:

$$
\overline{\pi}_t^i = \frac{\pi_{t-1}^i p\left(\mathbf{e}_t \mid \mathbf{s}_t^i\right)}{\sum_{j=1}^{N} \pi_{t-1}^j p\left(\mathbf{e}_t \mid \mathbf{s}_t^j\right)}. \tag{21}
$$

However, this choice has several drawbacks derived from the fact that not incorporating the observations introduces errors in the prediction. Thus, it may be the case that only a few particles have significant weights after being evaluated, specially when the likelihood function is much narrower than the temporal prior.

5.1 Degeneracy Problem

The SIS algorithm have an intrinsic problem which prevents from using it as it is. As it is proved in [5], the variance of the importance weights increase over time. This result has devastating consequences on the simulation performance, since the majority of the normalised importance weights tend to zero after few iterations. This samples being numerically insignificant, they are not taken into account in the pdf approximation. This result implies a sample wastage and a poor representation of the posterior distribution.

6 Sequential Importance Re-sampling (SIR)

Under this approach, a re-sampling stage is used to prune those particles with negligible importance weights, and multiply those with higher ones. Thus, samples are re-sampled with replacement using the importance weights as probabilities.

This idea is based on the *factored sampling* algorithm [9] designed for stationary pdf's. It works as follows: A posterior representation is given by the Bayes' theorem:

$$P(\mathbf{S} \mid \mathbf{e}) \propto P(\mathbf{e} \mid \mathbf{S}) P(\mathbf{S}), \tag{22}$$

but the likelihood function is complex enough to prevent the posterior being evaluated in closed form. Thus, sampling techniques are proposed to generate random variates from a distribution $\tilde{P}(\mathbf{s})$ that approximates the posterior $P(\mathbf{S} \mid \mathbf{e})$. A sample set of N i.i.d. random samples, $\left\{\hat{\mathbf{s}}^i; i = 1 : N\right\}$, is simulated from the initial prior density function, $P(\mathbf{S})$. The algorithm assigns normalised weights $\overline{\pi}^i$ to each sample in the set according to the likelihood function:

$$\overline{\pi}^i = \frac{p\left(\mathbf{e} \mid \widehat{\mathbf{s}}^i\right)}{\sum_{j=1}^{N} p\left(\mathbf{e} \mid \widehat{\mathbf{s}}^j\right)}. \tag{23}$$

Subsequently, the samples are selected —or re-sampled— from the sample set with probability $\overline{\pi}^i$. Therefore, the new sample set, $\left\{\mathbf{s}^i; i = 1 : N\right\}$, represents the posterior density function, $P(\mathbf{S} \mid \mathbf{e})$, accurately as $N \to \infty$. Obviously, some particles may be chosen several times, especially those with higher weights. Thus, some samples in the new set could be identical. On the other hand, samples with lower weights could be not chosen at all.

This weighted particle representation is shown in Fig. 2, where the posterior density function is represented by blobs whose centres are the sample set $\left\{\mathbf{s}^i; i = 1 : N\right\}$ and their area is proportional to the observation value given by the weights $\overline{\pi}^i$.

This idea was introduced by Gordon et al. [8] within a Bayesian filtering framework, thereby leading to *Sequential Importance Re-sampling* (SIR) filters. Here, a posterior probability density function represented by samples is iteratively computed. The pdf undergoes a diffusion-reinforcement process, and the reinforcement stage is followed by a run of the factored sampling algorithm presented above. Thus, the factored sampling is extended by applying it iteratively to successive time-slices.

Subsequently, this techniques were introduced in the Computer Vision field, as well as in other areas such as Artificial Intelligence, or Automatic Control. Therefore, these methods are also variously called: *particle filtering* —after the use of samples or particles as the way of propagating the probability density function— *survival of the fittest* —after the re-sampling stage— *bootstrap*

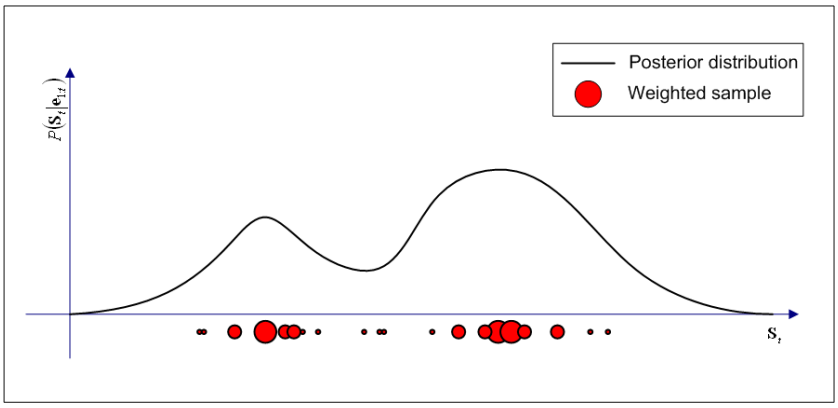

Figure 2: Posterior pdf representation as set of weighted particles. See text for details.

filtering[5], etc. In Computer Vision they are widely used under the name of CONDENSATION, after the paper presented in [10].

6.1 The CONDENSATION Algorithm

The CONDENSATION algorithm was presented by Isard and Blake in short form at the European Conference on Computer Vision in 1996 [10]. Later on, it was fully developed in [11]. This intended to track a human contour, which moves in cluttered background, given a raw video signal as data.

CONDENSATION addresses the filtering problem when no assumption about linearity or Gaussianity is made on almost all involved probability density functions. The algorithm is based on Bayesian filters. Therefore, it computes a posterior probability density function $P(\mathbf{S}_t \mid \mathbf{e}_{1:t})$ which undergoes the diffusion-reinforcement process described above. Because of the analytical problems already exposed, it makes use of Monte-Carlo simulation techniques.

It follows the aforementioned SIR approach. Thus, the posterior pdf at time $t-1$, $P(\mathbf{S}_{t-1} \mid \mathbf{e}_{1:t-1})$, is given by a set of tuples, each of them consisting in one sample and its weight, $\left\{\hat{\mathbf{s}}_{t-1}^i, \overline{\pi}_{t-1}^i; i = 1:N\right\}$ or, after applying the factored sampling algorithm, by the re-sampled sample set $\left\{\mathbf{s}_{t-1}^i, \frac{1}{N}; i = 1:N\right\}$. In this case, since all particles are evenly weighted, weights are not displayed and the notation is reduced to $\left\{\mathbf{s}_{t-1}^i; i = 1:N\right\}$.

Summarising, the four density functions involved in a Bayesian filter are:

1. the initial prior density function, $P(\mathbf{S}_0)$;
2. the transition model, $P(\mathbf{S}_t \mid \mathbf{S}_{t-1})$ for $t \geq 1$;

[5]The use of the term bootstrap derives from the phrase *"to pull oneself up by one's bootstrap"*, widely thought to be based on one of the eighteenth century *Adventures of Baron Munchausen*, by Rudolph Erich Raspe. In the context of this thesis, it means that the algorithm starts up and recovers by itself: fittest old samples give rise to many new ones.

3. the likelihood function, $P(\mathbf{E}_t \mid \mathbf{S}_t)$ for $t \geq 1$;

4. the posterior state density function, $P(\mathbf{S}_t \mid \mathbf{e}_{1:t})$ for $t \geq 1$.

The initial prior density function is now the only one supposed to be Gaussian. Therefore, the initial sampling is straightforward. Samples are propagated using the approach described above, that is, by sampling them from the transition model. Thus, there is no need to sample from the previous posterior in subsequently iterations. This fact avoids one of the main problems of the approach based on Monte Carlo Simulation, i.e., sampling from a complex, multivariate and only known up to a proportionality constant posterior pdf.

This algorithm works as follows: each iteration starts with the prediction stage where the temporal prior $P(\mathbf{S}_t \mid \mathbf{e}_{1:t-1})$ is obtained by applying the transition model $P(\mathbf{S}_t \mid \mathbf{S}_{t-1})$ to the previous posterior. Computationally, this is done in two steps. In the first place, a deterministic drift is applied to each sample of the previous posterior, $\left\{\mathbf{s}_{t-1}^i; i = 1:N\right\}$. Obviously, those samples which were identical will undergo the same drift. Then, the random component, i.e. the diffusion, is applied causing identical samples to split. As a result of this stage, the sample set represents the prior density function at time t, $\left\{\hat{\mathbf{s}}_t^i; i = 1:N\right\}$.

The second stage consists in the likelihood correction where the sample weights are calculated according to:

$$\pi_t^i = p\left(\mathbf{e}_t^i \mid \hat{\mathbf{s}}_t^i\right). \tag{24}$$

It is worth to notice that there is no need to recursively propagate the weights —as done in Eq. (21)— since all previous weights are even and equal to $\frac{1}{N}$ after the re-sampling stage. Once all samples have been propagated and measured, the final stage applies the factored sampling to carry out the re-sampling phase. Thus, weights are normalised:

$$\overline{\pi}_t^i = \frac{\pi_t^i}{\sum_{i=j}^{N} \pi_t^j}, \tag{25}$$

where $\overline{\pi}_t^i$ denotes the i-th sample normalised weight at time t.

Sampling from the discrete set $\left\{\hat{\mathbf{s}}_t^i; i = 1:N\right\}$ with probabilities $\overline{\pi}_t^i$ can be accomplished by sampling from a discrete uniform distribution, projecting the index onto the sample cumulative distribution range and then onto the distribution domain [5], see Fig. 3.

The cumulative probability distribution is constructed according to:

$$\begin{aligned} c_t^0 &= 0, \\ c_t^i &= c_t^{i-1} + \overline{\pi}_t^i, \quad i = 1:N. \end{aligned} \tag{26}$$

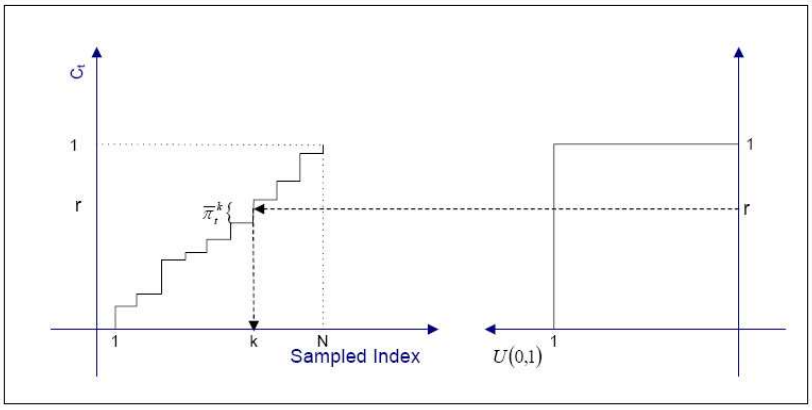

Figure 3: Cumulative distribution.

Algorithm 1 Re-sampling stage.

- **For** each sample $\mathbf{s}_t^i$:
  1. a random number is generated from a Uniform distribution, $r \in [0, 1]$.
  2. the smallest k index for which $c_t^k \geq r$ is found.
  3. the corresponding sample is selected, $\mathbf{s}_t^i = \hat{\mathbf{s}}_t^k$.
- **end for** i

Then, the new sample set, $\left\{\mathbf{s}_t^i; i = 1 : N\right\}$ is calculated by generating a random number, and selecting the sample whose corresponding cumulative probability exceed this number. This process is summarised in Algorithm 1.

Finally, the sample set represents the posterior pdf at time t, $P\left(\mathbf{s}_t, \mathbf{e}_{1:t}\right)$. The sample set size N is kept constant over time for all iterations. The expected value at time t can be approximated as:

$$\mathbb{E}_{P(\mathbf{S}_t|e_{1:t})}\left[\mathbf{S}_t\right] \approx \sum_{i=1}^{N} \overline{\pi}_t^i \hat{\mathbf{s}}_t^i \tag{27}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \mathbf{s}_t^i. \tag{28}$$

It is interesting to remark that the accuracy of any estimate —such as the mean and covariance— of the posterior distribution can only decrease as a result of the re-sampling stage. Thus, if these quantities are to be used or displayed, then these should be computed prior to re-sampling, as in Eq. (27), instead of using the posterior expression in Eq. (28).

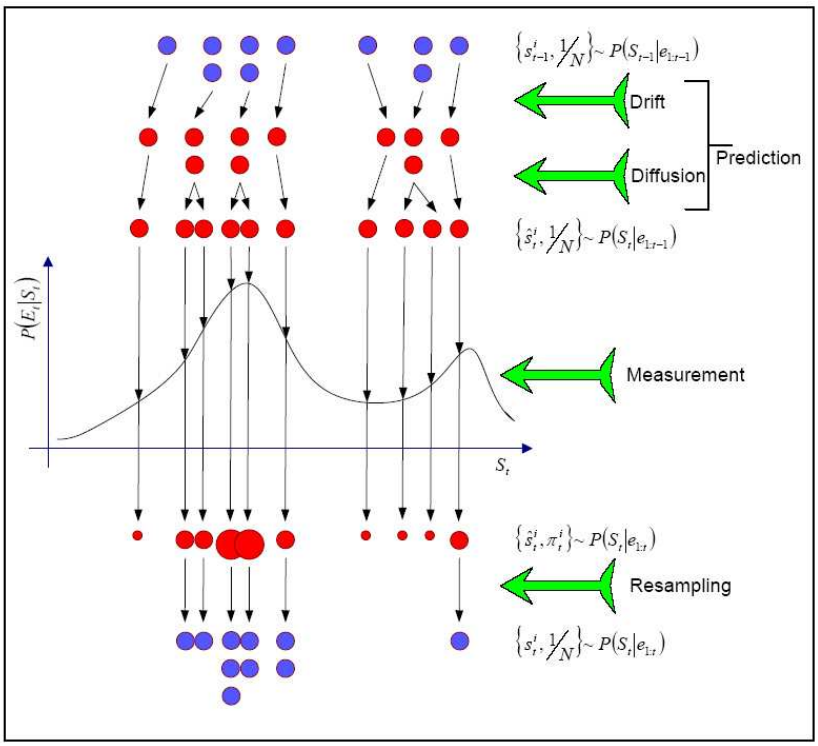

Figure 4: CONDENSATION algorithm: a graphical representation of one iteration. See text for details.

The algorithm is graphically depicted in Fig. 4, and summed up in Algorithm 2.

6.2 The Drawbacks of the CONDENSATION Algorithm

CONDENSATION has certainly been widely applied between 1999 and 2003. According to Cite-Seer[6], it has a peak of over 35 citations in 2001 and 271 hits within the Cite-Seer database. It has been considered fast and efficient due to its two main advantages:

1. first of all, it can represent multi-modal density functions. This fact allows us to consider multiple hypotheses, which is essential in scenes where background clutter or other moving objects[7] could mimic the target. Thus, it is possible to propagate multiple hypotheses which are pruned or reinforced in each iteration depending on their likelihood.

[6] http://citeseer.ist.psu.edu/

[7] Which does not mean that several targets can be tracked at the same time using the algorithm as it is.

Algorithm 2 CONDENSATION.

PROPAGATION

- **for** each sample in the set $\left\{\mathbf{s}_{t-1}^i; i = 1 : N\right\}$ **do**
  1. predict the sample values $\hat{\mathbf{s}}_t^i$ using the transition model $P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right)$;
  2. measure the sample weights π_t^i, Eq. (24);
- **end for** i

STATE ESTIMATION

- Estimate the state according to Eq. (27);

RE-SAMPLING

- Normalise the weights, Eq. (25);
- Compute the cumulative probabilities as in Eq.(26);
- **Call** the algorithm in Algorithm 1.

2. The second advantage is that, maintaining the sample set size fixed, it was supposed to be able to run with bounded computational resources in near real time[8].

Isard and Blake proved in [11] the asymptotic correctness of the algorithm by showing that the sample set representation of the posterior density function has weak and uniform convergence as $N \rightarrow \infty$. Thus, it is stated that each sample at time t of the sample set $\left\{\mathbf{s}_t^i; i = 1 : N\right\}$ is drawn from a probability density function $\widetilde{P}\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right)$ such that $\widetilde{P}\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right) \rightarrow P\left(\mathbf{S}_t \mid \mathbf{e}_{1:t}\right)$, where $\rightarrow$ denotes weak, uniform convergence[9].

However, they already warned that the convergence was proved for $N \rightarrow \infty$ *given a fixed* t. Therefore, the sampled representation approximates the true distribution with a desired accuracy but only for a fixed number of frames T. Nothing is said about the limit $T \rightarrow \infty$. Thus, *at later times larger values of* N *may be required.*

[8] However, as will be shown later, having a fixed sample set size has several drawbacks. Further, the number of samples required to ensured acceptable performances in high dimensional spaces prevent from a real-time use in most applications. An on-line sample-set size adaptation was explore was Fox [7] by evaluating the approximation error using the Kullback-Leibler distance; this was kept bounded by modifying the sample set size.

[9] **Weak convergence:** for every Q defined in a probability space, $\left\langle \widetilde{P}\left(\mathbf{s}_t \mid \mathbf{e}_{1:t}\right), Q \right\rangle \rightarrow \left\langle P\left(\mathbf{s}_t \mid \mathbf{e}_{1:t}\right), Q \right\rangle$ where $\langle\rangle$ denotes the inner product.
Uniform convergence: for every $\varepsilon > 0$, there exists a natural number N such that for all $\mathbf{s}_t$ and all $n > N$, $\left|\widetilde{p}\left(\mathbf{s}_t \mid \mathbf{e}_{1:t}\right) - p\left(\mathbf{s}_t \mid \mathbf{e}_{1:t}\right)\right| < \varepsilon$.

They also stated that *there is no information about how large N should be* for a requested precision and, therefore, it is heuristically determined. These and other undesirable CONDENSATION side-effects were thoroughly discussed by King and Forsyth [15]. They are briefly presented in the next paragraphs.

One of the main drawbacks of the re-sampling algorithms is a phenomenon called *sampling impoverishment*. Let us consider that the samples are spread around several *modes*[10]. King and Forsyth demonstrated that, with probability one —what is called an *almost sure* event[11]—, all samples will end up in one of those modes. Moreover, the probability that one mode absorbs all samples is proportional to the number of samples that started in it. Therefore, spurious modes have a non-zero probability of usurping all samples, causing the true mode to be lost.

Although sampling impoverishment is well studied and proved in [15], it can also be informally explained as a result of what is called *genetic drift*: consider a finite population and one particular gene. The frequency of the gene will not be exactly reproduced in the offspring due to sampling errors. This sampling error is propagated over time. The initial frequency is lost because there is not any kind of genetic memory. Eventually, this random process leads to a population where this gene is either lost or is present in every individual. In both cases, no further changes are possible. Thus, one mode has disappeared and it cannot be recovered. The Markov chain that modelled the process has reached an *absorbing state*, and its distribution is known as a *stationary distribution* which means that $P(\mathbf{S}_{t+1}) = P(\mathbf{S}_t)$.

CONDENSATION uses factored sampling. This process involves a loss of information. The probability for one sample of being selected is given by its weight. Consider now that several samples could be identical and similar samples form modes that can be far enough one from the other. The probability of propagating one mode is proportional to the number of samples that constitute it. Sample impoverishment means that all but one of these modes could disappear, and this fact has a non-negligible probability of happening in finite time.

Considering a real-time tracking application —whose frame rate can be set for instance at 30 frames per second, which means 30 generations per second— it is obvious that many modes could disappear in less than seconds. How many seconds will be needed is only a matter of how many samples are used.

Moreover, lost modes have a very low probability of being recovered. The diffusion process could preserve diversity, as mutation does in genetics. However, the distance between modes is usually bigger than the diffusion. One sample will need several iterations in order to move from one mode to another. But the likelihood in the region between modes is small, thereby making such a journey highly improbable.

[10] The term mode here refers to each local maximum of the distribution.

[11] There is a subtle difference between an event being *sure* and *almost sure*. On the one hand, a sure event will always happen, and no other event can ever happen. On the other, if an event is almost sure, other event are allowed to occur, but they happen almost never. Thus, for instance, infinite sequences of events, or a continuum of outcomes, allow events with zero-probability to occur —like hitting with a dart a particular point.

Summarising, *there is a non-negligible probability of losing modes, a low probability of recovering them, and the remaining modes could be all spurious.*

There is also another interesting fact, albeit undesirable as well. Isolated populations, starting with identical gene frequency, can end up in different absorbing states. Thus, variation within populations is turned into variations between populations. Returning to the tracking problem, this fact means that different runs of the algorithm lead to different results. Therefore, *computed expectations may have high variance*. However, *computed expectations within the same algorithm run have low variance* making the tracker look stable.

A yet another remarkable phenomenon is caused by the tendency of CONDENSATION towards clustering samples. *Even when the likelihood function gives no information at all*, i.e, there is nothing to track in the scene, *samples become quickly concentrated*. It strongly looks as if the tracker is following something, when actually it isn't. Of course, the peaks tracked differ from run to run.

Finally, CONDENSATION was designed to keep multiple hypotheses but only for a single target. Thus, multiple-target tracking was not feasible. Further extensions and variations from other authors [20, 16] usually lead to the so-called *curse of dimensionality*[12].

King and Forsyth proposed two approaches to tackle sampling impoverishment. In the first place, they suggested using fewer re-sampling steps. Obviously, a well constrained dynamic model would be required, what is usually not feasible. The second suggestion implies generating new samples occasionally. This suggestions has been followed by Varona et al. in [26], and within the importance-sampling framework, by Isard and Blake [12].

7 An Approach to MTT by Particle Filtering

In this section, an proposal based on particle filters is developed in order to perform Multiple-Target Tracking. The approach was initially inspired in the *iTrack* algorithm —within the SIR framework— implemented by Varona in his PhD thesis [25]. Subsequently, the focus has been placed in coping with two main difficulties:

1. inherent drawbacks of SIR methods;
2. and, scenario-dependent problems.

On the one hand, serious computational problems arose due to the inability of managing particle sets which must be big enough to populate adequately the search space, thereby being able of representing arbitrary distributions. Thus, particles should be wisely steered and re-sampled, so as to reduce the number

[12] This is a term coined by Richard Bellman in 1961 to refer to the problem caused by the exponential increase of an hyper-volume as a function of space dimensionality: adding extra dimensions causes an exponential growth of the number of required samples to densely populate the space.

of required particles. Issues such as sample impoverishment, and the curse of dimensionality must be tackle in a principled way.

On the other hand, robust tracking requires to deal with expected difficulties, such as background clutter and target occlusion. The non-rigid nature of the targets, along with changing illumination conditions, make model updating unavoidable. However, model drift should be prevented at any cost to ensure tracking viability.

7.1 State Modelling

A first-order dynamic model in image coordinates is used to model the motion of the central point of a bounding box. This bounding box is considered the region within the scene which is thought to enclose the target.

Thus, the target's motion is characterised by its position at time t, $\mathbf{x}_t = (x_t, y_t)^T$, and its speed, $\mathbf{u}_t = (u_t, v_t)^T$. This dynamic model involves the assumption of constant speed —acceleration will be given by Gaussian noise—- which can be more o less realistic depending on the target's dynamics and the frame rate. It usually holds in trajectory-analysis applications at current common frame rates of 25-30 fps.

The aspect model is given by a bounding box and an appearance matrix. The former, denoted by $\mathbf{w}_t = (w_t, h_t)^T$, defines a rectangle whose size is given by its width, w_t, and its height, h_t. The latter, denoted by $\mathbf{A}_t$, stores the pixel intensity values within the bounding box. An indicator of the expected likelihood value is given by λ_t. This stores expected matching, taking into account that differences will be found due to sensor noise, changes in illumination, shape deformations, etc.

The occlusion status is inferred and store in ρ_t. This is a binary variable which points out whether the target is the nearer one in a group to the camera.

Finally, a label l associates a specific appearance model to the corresponding samples, allowing multiple-target tracking. Therefore, the $l-$target's state is defined as $\mathbf{s}_t^l = \left(\mathbf{x}_t^l, \mathbf{u}_t^l, \mathbf{w}_t^l, \mathbf{A}_t^l, \rho_t^l, \lambda_t^l\right)^T$.

7.2 Transition Model

Several independence relationships are assumed in order to determine the transition model. It is considered that both aspect and dynamic models are independent, that the position only depends on the previous position and speed, the speed on the previous one, and so does the bounding box on and the appearance. Therefore, the transition model can be split:

$$\begin{aligned} P\left(\mathbf{S}_t \mid \mathbf{S}_{t-1}\right) &= P\left(\mathbf{X}_t, \mathbf{U}_t, \mathbf{W}_t, \mathbf{A}_t \mid \mathbf{X}_{t-1}, \mathbf{U}_{t-1}, \mathbf{W}_{t-1}, \mathbf{A}_{t-1}\right) \\ &= P\left(\mathbf{X}_t \mid \mathbf{X}_{t-1}, \mathbf{U}_{t-1}\right) P\left(\mathbf{U}_t \mid \mathbf{U}_{t-1}\right) P\left(\mathbf{W}_t \mid \mathbf{W}_{t-1}\right) P\left(\mathbf{A}_t \mid \mathbf{A}_{t-1}\right). \end{aligned} \tag{29}$$

Given the constant speed assumption, the dynamic model can be defined according to:

$$P(\mathbf{X}_t \mid \mathbf{x}_{t-1}, \mathbf{u}_{t-1}) = \mathcal{N}(\mathbf{X}_t; \mathbf{x}_{t-1} + \mathbf{u}_{t-1}\Delta_t, \mathbf{\Sigma_x}), \quad (30)$$

$$P(\mathbf{U}_t \mid \mathbf{u}_{t-1}) = \mathcal{N}(\mathbf{U}_t; \mathbf{u}_{t-1}, \mathbf{\Sigma_u}). \quad (31)$$

Thus, the position state variable $\mathbf{X}_t$ evolves according to a linear Gaussian whose mean is a linear expression of its parents and the variance is fixed and heuristically determined. Δ_t is the sampling period. Time is considered discrete and measured in frames. Thus, Δ_t equals 1. Position is also discrete and measured in pixels. On the other hand, the speed state variable $\mathbf{U}_t$ evolves according to a Gaussian whose mean is its parent and the variance is again heuristically fixed according to the expected target acceleration. These two covariance matrices are denoted by $\mathbf{\Sigma_x}$ and $\mathbf{\Sigma_u}$.

In order to implement the aspect model, it is assumed that the shape evolves smoothly, and the appearance is fixed between consecutive frames according to:

$$P(\mathbf{W}_t \mid \mathbf{w}_{t-1}) = \mathcal{N}(\mathbf{W}_t; \mathbf{w}_{t-1}, \mathbf{\Sigma_w}), \quad (32)$$

$$P(\mathbf{A}_t \mid \mathbf{A}_{t-1}) = \delta(\mathbf{A}_t - \mathbf{A}_{t-1}). \quad (33)$$

where $\mathbf{\Sigma_w}$ denotes the size covariance matrix.

Although the appearance is considered to be fixed when propagating the state, it will eventually be updated once the posterior expectation is computed.

Therefore, the position, speed, and size of each sample are predicted according to:

$$\begin{aligned} \hat{\mathbf{x}}_t^{i,l} &= \mathbf{x}_{t-1}^{i,l} + \mathbf{u}_{t-1}^{i,l}\Delta_t + \xi_{\mathbf{x}}^i, \\ \hat{\mathbf{u}}_t^{i,l} &= \mathbf{u}_{t-1}^{l} + \xi_{\mathbf{u}}^i, \\ \hat{\mathbf{w}}_t^{i,l} &= \mathbf{w}_{t-1}^{i,l} + \xi_{\mathbf{w}}^i, \end{aligned} \quad (34)$$

where the random vectors $\xi_{\mathbf{x}}^i, \xi_{\mathbf{u}}^i, \xi_{\mathbf{w}}^i$, sampled from WAGN processes, provide the system with a diversity of hypotheses.

Sample likelihoods depend on sample position and size, but not on their speeds. Thus, if speeds were propagated considering the previous speed, they would be in quasi open loop[13]. Thus, their values could become completely different from the true values within a few frames, and an important proportion of samples would be wasted. In order to avoid this phenomenon, the estimated target speed $\mathbf{u}_{t-1}^l$ at time $t-1$ is fed back into the prediction of $\hat{\mathbf{x}}_t^{i,l}$.

After the initialisation, no sample is generated using detection, since it would mask tracking misbehaviours. Thus, just tracking performances are tested by means of propagating hypotheses and weighting them according to evidence. Clearly, by incorporating detection, the general performance will be enhanced, providing the system with error-recovery capabilities.

[13] There would still be a weak relation, since speeds are used to predict positions, and position errors can be measured, but a considerable delay would be introduced, as it will be shown in the experimental results.

7.3 Template-based Likelihood Function

In a visual tracking context, the likelihood function gives the probability density function of image features given the state. The intensity is chosen as image feature. Features are considered pixel-oriented. Hence, the appearance is given by a matrix whose elements are the pixels' intensity values.

Let $\mathbf{I}_t$ be a matrix whose elements are the scene pixel intensity values at time t. Thus, evidence $\mathbf{e}_t$ is given by the input image sequence $\mathbf{I}_t$. Given the predicted position $\mathbf{X}_t$ and bounding-box size $\mathbf{W}_t$, the corresponding image sub-region is denoted by $\mathbf{I}_t^p$. The model appearance matrix must be scaled according to the sample size. Let $\mathbf{A}^s$ be the model scaled matrix. Thus, assuming that the likelihood function is independent of the speed component, it can be expressed as:

$$\begin{aligned} P\left(\mathbf{I}_t \mid \mathbf{S}_t\right) &= P\left(\mathbf{I}_t \mid \mathbf{X}_t, \mathbf{W}_t, \mathbf{A}_t\right) \\ &= P\left(\mathbf{I}_t^p \mid \mathbf{A}_t^s\right), \end{aligned} \tag{35}$$

and, once assumed constant appearance between frames and *White Additive Gaussian Noise*, the likelihood function can be defined as a similarity measure which averages the likelihood of all pixels within the bounding box[14]:

$$\begin{aligned} P\left(\mathbf{I}_t^p \mid \mathbf{A}_t^s\right) &= \frac{1}{M} \sum_{a,b \in \mathbf{A_t^s}} P\left(\mathbf{I}_t^p\left(a,b\right) \mid \mathbf{A}_t^s\left(a,b\right)\right) \\ &= \frac{1}{M} \sum_{a,b \in \mathbf{A_t^s}} \mathcal{N}\left(\mathbf{I}_t^p\left(a,b\right); \mathbf{A}_t^s\left(a,b\right), \sigma_n^2\right), \end{aligned} \tag{36}$$

where M is the number of pixels of the appearance model, (a,b) defines a pixel position in the appearance matrix and σ_n^2 is the camera noise variance, which randomly influences the pixels' intensity values.

7.4 Weight Normalisation

In a multiple-target tracking scenario, those targets whose samples exhibit lower likelihood are more likely to be lost, since the probability of propagating one mode is proportional to the cumulative weights of its samples. In order to avoid one target absorbing other target samples, genetic drift must be prevented. Thus, a memory term, which takes into account the number of targets being tracked, is included. Weights are normalised according to:

[14] This expression does not pretend to follow a probabilistic derivation. The likelihood function is usually defined in terms of a *distance*, and this distance is here computed from the likelihood of each pixel within the bounding box.

$$\overline{\pi}_t^{i,l} = \frac{\pi_t^{i,l}}{\sum_{i=1,j=l}^{N} \pi_t^{i,j}} \frac{1}{L}, \tag{37}$$

where L is the number of tracked targets. Each weight is normalised according to the total weight of the target's samples. Thus, all targets have the same probability of being propagated, since the addition of the weights of each target samples sums $\frac{1}{L}$. This allows multiple-target tracking using a single PF framework, despite the differences between their likelihoods and the genetic drift phenomenon.

7.5 State Estimation

The l-target estimates are computed according to:

$$\begin{aligned}
\mathbf{x}_t^l &= (1-\alpha_{\mathbf{x}}) \left(\mathbf{x}_{t-1}^l + \mathbf{u}_{t-1}^l \Delta_t \right) + \alpha_{\mathbf{x}} \left(L \sum_{i=1}^{N} \overline{\pi}_t^{i,l} \hat{\mathbf{x}}_t^{i,l} \right), \\
\mathbf{u}_t^l &= (1-\alpha_{\mathbf{u}}) \, \mathbf{u}_{t-1}^l + \alpha_{\mathbf{u}} \left(\frac{\mathbf{x}_t^l - \mathbf{x}_{t-1}^l}{\Delta_t} \right), \\
\mathbf{w}_t^l &= (1-\alpha_{\mathbf{w}}) \, \mathbf{w}_{t-1}^l + \alpha_{\mathbf{w}} \left(L \sum_{i=1}^{N} \overline{\pi}_t^{i,l} \widehat{\mathbf{w}}_t^{i,l} \right),
\end{aligned} \tag{38}$$

where $\alpha_{\mathbf{x}}, \alpha_{\mathbf{u}}, \alpha_{\mathbf{w}} \in [0,1]$ denote the adaptation rates. Target speeds are not estimated according to sample speeds and their weights, since significant errors would be introduced: samples are chosen only because of sample weights, which do not directly depend on the current speed. This fact could imply a significant amount of jitter and many samples would be wasted. Therefore, target speeds are computed from successive position estimates. Further, both position and speed estimates are enhanced by regularising them according to their histories.

The target appearance must also be updated. However, this is a sensitive task which may lead to the well-known *model drift* phenomenon. Thus, models are then only updated when two conditions hold:

- the target is not occluded;
- and, the likelihood of the estimated target's state suggests that the estimate is sufficiently reliable.

In this case, they are updated using an adaptive filter:

$$\mathbf{A}_t^l = (1-\alpha_{\mathbf{A}}) \, \mathbf{A}_{t-1}^{l,s} + \alpha_{\mathbf{A}} \mathbf{I}_t^l, \tag{39}$$

where $\alpha_{\mathbf{A}} \in [0,1]$ is the learning rate, and $\mathbf{I}_t^l$ is the image sub-region cropped given the target new estimate position and size $\mathbf{x}_t^l, \mathbf{w}_t^l$.

In order to determine when the estimate is reliable, the likelihood of the current estimate is computed, $p\left(\mathbf{e}_t \mid \mathbf{s}_t^l\right)$. The appearance is then updated when this value is higher than an indicator of the expected likelihood value, calculated following an adaptive rule:

$$\lambda_t^l = (1 - \alpha_l)\, \lambda_{t-1}^l + \alpha_l p\left(\mathbf{e}_t \mid \mathbf{s}_t^l\right). \tag{40}$$

7.6 Occlusion handling

Although the appearance model is not updated during occlusions, these still constitute a main cause of catastrophic failures. Partial occlusions may cause inaccurate size updating, according to the area that can be seen. In case of complete occlusions, sample likelihoods are meaningless, and the re-sampling phase randomly propagate them, quickly losing the target.

Hence, proper handling of occlusions is crucial. The state binary variable ρ_t^l tracks the occlusion status. Occlusions are predicted according to the learnt dynamics. When the predicted occlusion is significant, and the target likelihood is lower than the expected one given by λ_t^l, the target state changes into occluded. Then, the following changes are introduced:

- neither the size, nor the velocity or the likelihood-expectation indicator are updated; the position is just propagated
- those samples belonging to the occluded target are not re-sampled. As a result, samples are spread around the target because of the uncertainty predictions terms. The other targets' samples are re-sampled, but are not assigned to the occluded target, since otherwise this one would monopolise the whole sample set.

When the occlusion is no longer predicted, or a sample likelihood exceeds the value previous to the occlusion, ρ_t^l turns into zero, which immediately implies pruning those samples with lower weights. Furthermore, all estimates are again updated.

7.7 Extension of the Tracking Algorithm

Bounding-boxes and templates can hardly model the shape and appearance of non-rigid targets. The target region representation is changed into an ellipse in order to reduce the number of background pixels included in the model. Now, the motion of the central point of an elliptical region is modelled using first-order dynamics in image coordinates.

Further, the target appearance is represented by means of colour histograms. Histograms are broadly used to represented human appearance, since they are claimed to be less sensitive than colour templates to rotations in depth, the camera point of view, non-rigid targets, and partial occlusions. By using colour as image feature instead of intensity, a better target disambiguation can be achieved.

Thus, the $l-$model is given by:

$$\overline{\mathbf{p}}^l = \left\{ p_k^l; k = 1:K \right\}, \tag{41}$$

where K is the number of bins, and the probability of each feature is:

$$p_k^l = C^l \sum_{a=1}^{M} \delta\left(b\left(\mathbf{x}_a\right) - k\right), \tag{42}$$

where C^l is a normalisation constant required to ensure that $\sum_{k=1}^{K} p_k^l = 1$, δ the Kronecker delta, $\{\mathbf{x}_a; a = 1:M\}$ the pixel locations, and $b\left(\mathbf{x}_a\right)$ a function that associates the given pixel to its corresponding histogram bin.

The l-labelled target's state is then defined as $\mathbf{s}_t^l = \left(\mathbf{x}_t^l, \mathbf{u}_t^l, \mathbf{w}_t^l, \overline{\mathbf{p}}^l, \rho_t^l, \lambda_t^l\right)^T$, where components are the ellipse position, velocity, both axes, the appearance model, the occlusion status, and the expected target likelihood.

7.7.1 A Colour-based Likelihood function

The target distribution at the predicted position $\hat{\mathbf{x}}_t^{i,l}$ and ellipse size $\hat{\mathbf{w}}_t^{i,l}$, is given by $\mathbf{p}_i^l$, which is calculated in the same way as the model. The similarity between two histograms can be computed using the following metric [2, 19]:

$$d_B = \sqrt{1 - \rho\left(\mathbf{p}, \overline{\mathbf{p}}^l\right)}, \tag{43}$$

where

$$\rho\left(\mathbf{p}, \overline{\mathbf{p}}^l\right) = \sum_{k=1}^{K} \sqrt{p_k \overline{p}_k^l}, \tag{44}$$

is known as the *Bhattacharyya coefficient*. Therefore, similar histograms have a high Bhattacharyya coefficient, which should correspond to high sample weights. The computed metric can be mapped using a Gaussian distribution [19], and samples are thus weighted according to:

$$\pi_t^{i,l} = p\left(\mathbf{e}_t \mid \hat{\mathbf{s}}_t^{i,l}\right) = \mathcal{N}\left(d_B; \mu, \sigma^2\right). \tag{45}$$

So far no background information has been used. However, tracking success depends on how distinguishable the target is from a local environment. Thus, foreground features present also in its surroundings should be less important for target localisation. Here, an approach similar to [2] is adopted by using a *centre-surround* model to compute the local background histogram $\mathbf{q}^l$ according to the outer region which encloses the target, see Fig. 5.

(a) (b)

Figure 5: Examples of a centre-surround model with safety margin. (a) Tracked van from a traffic-monitoring sequence. (b) Tracked person from an indoor surveillance application in a shopping centre. Regions from centre to border: target estimation, safety margin, surrounding background, and non-local background.

The local background region is given by an ellipse which encloses the tracked one by defining two margins of dimension $\kappa_s * \max(h, w)$. The potential incorporation of own target pixels, specially if the target shape cannot be fairly represented by an ellipse is minimised by taking into account just the outer region to build the local background histogram. κ_s is usually equal to 0.1 for the inner margin and 0.3 for the outer one. Hence, the background histogram is used to compute a weight for each bin:

$$\omega_k^l = \left\{ \min \left(\frac{q_k^{l*}}{q_k^l} \right); k = 1 : K \right\}, \tag{46}$$

where q_k^{l*} is the minimum non-zero value. Thus, these weights are then applied to the target histogram to diminish the importance of those bins which represent the local background. Hence, the resulting Bhattacharyya coefficient is

$$\rho_w \left(\mathbf{p}, \overline{\mathbf{p}}^l \right) = \sum_{k=1}^{K} \omega_k^l \sqrt{p_k \overline{p}_k^l}. \tag{47}$$

Finally, in the state-estimation stage, Eq. (39) is changed accordingly:

$$\overline{\mathbf{p}}_t^l = (1 - \alpha_{\mathbf{q}}) \, \overline{\mathbf{p}}_{t-1}^l + \alpha_{\mathbf{q}} \mathbf{p}_t^l, \tag{48}$$

where $\alpha_{\mathbf{q}} \in [0, 1]$ is the learning rate which weights the most recent values versus the historic ones. The complete algorithm is summarised in Algorithm 3.

Algorithm 3 MTT particle filtering

PROPAGATION

- **for** $i = 1$ **to** N **do**
  1. predict the sample values $\hat{\mathbf{s}}_t^{i,l}$ using the transition model in Eq. (34)
  2. measure the sample weights π_t^i according to Eq. (45)
- **end for** i

UPDATING

- normalise the weights as in Eq. (37)
- predict occlusion percentage according to target's dynamics models
- **for** $l = 1$ **to** L **do**
  1. evaluate occlusions according to target collision and likelihoods
  2. estimate the target state:
     (a) **if** target is occluded **then** set adaptation rates $\alpha_{\mathbf{x}}, \alpha_{\mathbf{u}}$ to zero
     (b) estimate target position and speed according to Eq. 38
     (c) **if** the target estimate is reliable
        i. update target's size
        ii. update the appearance models following Eqs. (38),(48)
        iii. update λ_t^l as in Eq.(40)
- **end for** l

RE-SAMPLING

- Build the cumulative distribution as in Eq.(26)
- **for** $i = 1$ **to** N **do**
  if target l is occluded **then** keep the sample: $\mathbf{s}_t^{i,l} = \hat{\mathbf{s}}_t^{i,l}$.
  else proceed with re-sampling as in Algorithm **1**
- **end for** i

8 Discussion

With this work we have attempted to take a step towards solving the numerous difficulties which appear in MTT applications by means of particle filtering.

Dynamics updating is modified by feeding back the estimated speed into

the prediction stage. The target's speed is estimated from successive position estimates. Both position and speed estimates are now regularised. Thus, sample wastage is significantly reduced. In addition, trajectory jitter is considerably attenuated.

Different likelihood function have been explored in order to properly evaluate samples associated to targets which present a high appearance variability. Finally, the approach relies on the Bhattacharyya coefficient between colour histograms to perform this task.

Model updating is carried out with special care, in order to overcome the model drift phenomenon. A multiple-target tracking scenario causes several problems, including sampling impoverishment and mutual occlusions. These issues are tackled by redefining the weight normalisation, and predicting and handling occlusions. The proposed sample-weight normalisation avoids losing any of the targets due to the lack of samples.

Although significant advances have been obtained the approach is far from being suitable to perform multiple target tracking in cluttered environments under uncontrolled conditions in long sequences. This is due to multiple facts:

- Monte-Carlo methods are usually not able to densely populate a high-dimension spaces. Estimations are performed from a limited number of samples. This results in poor state approximations when dealing with multi-modal pdf's.
- Top-down approaches require extremely constrained models, which is not feasible in generic applications. Errors in the estimation are propagated, thereby causing model drift.
- An independent observation process from prediction is required to cope with estimation errors with a finite number of samples. This entails the necessity a bottom-up process.
- Likelihood functions are usually not discriminative enough.

As stated by the English Franciscan Friar William of Ockham in the 14th century, "entia non sunt multiplicanda praeter necessitatem". This principle[15] suggests to select the theory that introduces the fewest assumptions and postulates the fewest entities, which is of course not the case of PF's in uncontrolled environments.

References

[1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on PFs for On-line Non-linear/Non-Gaussian Bayesian Tracking. *Signal Processing*, 50(2):174–188, 2002. (Cited on page 1)

[15] It is usually referred as the 'Ockham´s razor'

[2] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based Object Tracking. *PAMI*, 25(5):564–577, 2003. (Cited on page 23)

[3] N. de Freitas, A. Gee M. Niranjan, and A. Doucet. Sequential Monte Carlo Methods for Optimisation of Neural Network Models. Technical Report TR 328, Cambridge University, 1998. (Cited on page 1)

[4] J. Deutscher and I. Reid. Articulated Body Motion Capture by Stochastic Search. *IJCV*, 61(2):185–205, 2005. (Cited on page 1)

[5] A. Doucet. On Sequential Simulation-Based Methods for Bayesian Filtering. Technical Report TR310, Cambridge University, 1998. (Cited on pages 1, 4, 9, and 12)

[6] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer-Verlang, first edition, 2001. (Cited on page 1)

[7] D. Fox. Adapting the Sample Sieze in Particle Filters Through KLD-Sampling. *Int. Journal of Robotics Research*, 22(12):985–1004, 2003. (Cited on page 15)

[8] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-gaussian Bayesian state estimation. In *IEE Proceedings-F*, volume 140, pages 107–113, 1993. (Cited on pages 1 and 10)

[9] U. Grenander, Y. Chow, and D. M. Keenan. *HANDS. A Pattern Theoretical Study of Biological Shapes*. Springer-Verlang, 1991. (Cited on page 10)

[10] M. Isard and A. Blake. Contour Tracking by Stochastic Propagation of Conditional Density. In *4th ECCV, Cambridge UK*, volume 1, pages 343–356. Springer-Verlang, 1996. (Cited on pages 1 and 11)

[11] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998. (Cited on pages 1, 9, 11, and 15)

[12] M. Isard and A. Blake. Icondensation Unifying Low-level and High-level Tracking in a Stochastic Framework. In *5th ECCV, Freiburg, Germany*, volume 1, pages 893–908, 1998. (Cited on pages 1 and 17)

[13] M. Isard and J. MacCormick. BraMBLe: A Bayesian Multiple-Blob Tracker. In *8th ICCV, Vancouver, Canada*, volume 2, pages 34–41. IEEE, 2001. (Cited on page 1)

[14] R. Kalman. A New Approach to Linear Filtering and Prediction Problems. *ASME–Journal of Basic Engineering*, 82(D):35–45, 1960. (Cited on page 1)

[15] O. King and D. Forsyth. How Does CONDENSATION Behave with a Finite Number of Samples? In *6th ECCV, Ireland*, volume 1, pages 695–709, 2000. (Cited on pages 1 and 16)

[16] J. MacCormick and A. Blake. A Probabilistic Exclusion Principle for Tracking Multiple Objects. *IJCV*, 39(1):57–71, 2000. (Cited on pages 1 and 17)

[17] J. MacCormick and M. Isard. Partioned Sampling, Articulated Objects, and Interface-quality Hand Tracking. In *6th ECCV, Dublin, Ireland*, volume 2, pages 3–19. Springer-Verlang, 2000. (Cited on page 1)

[18] D. Mackay. *Introduction to Monte Carlo Methods*, chapter 7, pages 175–204. MIT Press, 1998. (Cited on page 1)

[19] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An Adaptive Color-Based Particle Filter. *Image and Vision Computing*, 21(1):99–110, 2003. (Cited on pages 1 and 23)

[20] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based Probabilistic Tracking. In *7th ECCV, Copenhaguen, Denmark*, pages 661–675. Springer-Verlang, 2002. (Cited on pages 1 and 17)

[21] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter*. Artech House Publising, first edition, 2004. (Cited on page 1)

[22] S. M. Ross. *Simulation*. Academic Press, 2nd edition, 1997. (Cited on page 1)

[23] R. Russell and P. Norvig. *Artificial Intelligence, a Modern Approach*, chapter 13-15. Prentice Hall, 2nd edition, 2003. (Cited on pages 1, 2, 3, and 4)

[24] R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan. The Unscented Particle Filter. Technical Report TR380, Cambridge University, 2000. (Cited on page 1)

[25] X. Varona. *Seguimiento Visual Robusto en Entornos Complejos*. PhD thesis, Universitat Autònoma de Barcelona, Spain, 2001 (in Spanish). (Cited on page 17)

[26] X. Varona, J. Gonzàlez, X. Roca, and J. Villanueva. iTrack: Image-based Probabilistic Tracking of People. In *15th ICPR, Barcelona, Spain*, volume 3, pages 1110–1113. IEEE, 2000. (Cited on pages 1 and 17)

[27] Y. Wu, T. Yu, and G. Hua. Tracking Appearances with Occlusions. In *CVPR, Wisconsin, USA*, volume 1, pages 789–795. IEEE, 2003. (Cited on page 1)