1

Visual Tracking by Particle Filtering

1.1. Introduction

The aim of this introductory chapter is to give a brief overview of the progress made over the last 20 years in visual tracking by particle filtering. To begin (section 1.2), we will present the theoretical elements necessary for understanding particle filtering. Thus, we will first introduce recursive Bayesian filtering, before giving the outline of particle filtering. For more details, in particular theorem demonstrations and convergence studies, we invite the reader to refer to more advanced studies [CHE 03b, DOU 00b, GOR 93]. We will then explain how particle filtering is used in visual tracking in video sequences. Although the literature is abundant on this subject and evolving very fast, it is impossible to give a complete overview of this subject. Next, section 1.3 presents certain limits of particle filtering. Toward the end, we specify our scientific position in section 1.4 and the methodological axes that allow a part of these problems to be solved. Finally, section 1.5 gives the current state of the main large families of approaches that are concerned with managing large-sized state and/or observation spaces in particle filtering.

1.2. Theoretical models

1.2.1. *Recursive Bayesian filtering*

Recursive Bayesian filtering [JAZ 70] aims to approximate the state of a hidden Markov process, which is observed through an observation equation. Let $\{\mathrm{x}_{0:t}\} = \{\mathrm{x}_0, \ldots, \mathrm{x}_t\}$ be this process, where x_t is the state vector, $\mathbf{y}_t$ the observation at instant t and the two models:

$$\begin{cases} \mathrm{x}_t = f_t(\mathrm{x}_{t-1}, \mathbf{u}_t) \\ \mathbf{y}_t = g_t(\mathrm{x}_t, \mathbf{v}_t) \end{cases} \qquad [1.1]$$

The first equation is the state equation, with the state transition function f_t between the instants $t-1$ and t, and the second is the observation equation, giving the measurement of the state through an observation function g_t. $\mathbf{u}_t$ and $\mathbf{v}_t$ are independent white noises.

All information necessary for approximating $\mathrm{x}_{0:t}$ is contained in the *a posteriori* density, also known as the filtering density, $p(\mathrm{x}_{0:t}|\mathbf{y}_{1:t})$, where $\mathbf{y}_{1:t} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t\}$, in which we can prove, by applying the definition of conditional probabilities, that it follows the following recursive equation for a known [CHE 03b] $t \geq 1$ ($p(x_0)$:

$$p(\mathrm{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathrm{x}_{0:t})p(\mathrm{x}_t|\mathrm{x}_{0:t-1}, \mathbf{y}_{1:t-1})p(\mathrm{x}_{0:t-1}|\mathbf{y}_{1:t-1})}{\int_{\mathrm{x}_{0:t}} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathrm{x}_{0:t})p(\mathrm{x}_{0:t}|\mathbf{y}_{1:t-1})d\mathrm{x}_{0:t}} \qquad [1.2]$$

Under the Markov hypothesis, $p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \mathrm{x}_{0:t}) = p(\mathbf{y}_t|\mathrm{x}_t)$ (the observations at different instants are independent between themselves given the states and do not depend on the state at the current instant) and

$p(\mathrm{x}_t|\mathrm{x}_{0:t-1}, \mathbf{y}_{1:t-1}) = p(\mathrm{x}_t|\mathrm{x}_{t-1})$ (the current state only depends on the previous state), equation [1.2] becomes:

$$p(\mathrm{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathrm{x}_t)p(\mathrm{x}_t|\mathrm{x}_{t-1})p(\mathrm{x}_{0:t-1}|\mathbf{y}_{1:t-1})}{\int_{\mathrm{x}_{0:t}} p(\mathbf{y}_t|\mathrm{x}_t)p(\mathrm{x}_{0:t}|\mathbf{y}_{1:t-1})d\mathrm{x}_{0:t}} \qquad [1.3]$$

The state transition equation is represented by the density $p(\mathrm{x}_t|\mathrm{x}_{t-1})$ and is linked to the f_t function. This density is also called the transition function and gives the probably state x_t at the instant t, given its previous state x_{t-1}. The observation equation is represented by $p(\mathbf{y}_t|\mathrm{x}_t)$ and is linked to the function g_t. This density is also called the likelihood function and gives the probability of making the observation $\mathbf{y}_t$ given the state x_t. We can see that equation [1.3] is recursive and it decomposes into two primary stages that we detail below.

1) The first stage, known as prediction step, allows approximating the *a posteriori* density $p(\mathrm{x}_{0:t}|\mathbf{y}_{1:t-1})$ using the transition distribution $p(\mathrm{x}_t|\mathrm{x}_{t-1})$ and the previously approximated density $p(\mathrm{x}_{0:t-1}|\mathbf{y}_{1:t-1})$.

2) The second stage, called correction step, allows obtaining the *a posteriori* density $p(\mathrm{x}_{0:t}|\mathbf{y}_{1:t})$, using the likelihood distribution $p(\mathbf{y}_t|\mathrm{x}_t)$, which depends on the new observation. This *a posteriori* density represents the density of the probability to have the set of states $\mathrm{x}_{0:t}$, among all the possible states, given the history of the observations $\mathbf{y}_{1:t}$.

In order to obtain calculable estimators of $\mathrm{x}_{0:t}$, we can use, for example, the conditional mean, given by:

$$\mathbb{E}_p[\mathcal{F}(\mathrm{x}_{0:t})] = \int_{\mathrm{x}_{0:t}} \mathcal{F}(\mathrm{x}_{0:t})p(\mathrm{x}_{0:t}|\mathbf{y}_{1:t})d\mathrm{x}_{0:t} \qquad [1.4]$$

where $\mathcal{F}$ is some bounded function. If the densities are Gaussian, then there exists a solution (analytical expression of the Gaussian parameters to approximate) given by the Kalman filter [KAL 60]. Otherwise, the whole of equation [1.4] is not calculable directly. We can invoke, under special conditions, the solutions given by the following types of methods:

– analytical (extended Kalman filter [JAZ 70], unscented Kalman filter [JUL 97]) that approach the law from a Gaussian sum and are better adapted to weakly nonlinear and unimodal cases, which is nonetheless not appropriate for most problems of vision;

– numerical (approximations by discrete tables, division into parts) that are, most of the time, complex to solve, not very flexible and only adapted to state spaces of a small size.

Most of the time, in vision, solutions are not adapted as the integrals are not directly calculable. For the general case (non-parametric and multi-modal densities), it is necessary to make use of numerical approximations, such as those provided by sequential Monte-Carlo methods, which we will present in the following section and that are the methodological heart of this work.

1.2.2. *Sequential Monte-Carlo methods*

Sequential Monte-Carlo methods, also known under the name of particle filters (PFs), were studied by many researchers at the beginning of the 1990s [GOR 93, MOR 95] and combine Monte-Carlo simulation and recursive Bayesian filtering. Today, they are widely used in the computer visualization community. Before detailing the principle of particle filtering, we need to introduce importance sampling.

1.2.2.1. *Importance sampling*

Once the *a posteriori* density defined by equation [1.3] has been approximated, we can evaluate the estimator given in equation [1.4]. The Monte-Carlo method allows us to approximate this integral with the realization of a random variable distributed according to the *a posteriori* density.

Unfortunately, we are almost never able to sample following this law, so to solve this problem, we introduce a proposal function (or importance function) $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, whose support contains $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ and from which we can sample. The conditional mean is then given by:

$$\begin{aligned}\mathbb{E}_p[\mathcal{F}(\mathbf{x}_{0:t})] &= \int_{\mathbf{x}_{0:t}} \mathcal{F}(\mathbf{x}_{0:t}) \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \\ &= \mathbb{E}_q \left[\mathcal{F}(\mathbf{x}_{0:t}) \frac{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})} \right] \qquad [1.5]\end{aligned}$$

With N realizations $\mathbf{x}_{0:t}^{(i)} \sim q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, $i = 1, \ldots, N$, we can approximate the previous estimator by:

$$\hat{\mathbb{E}}_p[\mathcal{F}(\mathbf{x}_{0:t})] = \frac{1}{N} \sum_{i=1}^{N} \mathcal{F}(\mathbf{x}_{0:t}^{(i)}) \frac{p(\mathbf{x}_{0:t}^{(i)})|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})} \qquad [1.6]$$

The law of large numbers allows us to show that this estimator almost certainly converges toward $\mathbb{E}_p[\mathcal{F}(\mathbf{x}_{0:t})]$ when N tends to infinity. Thus, we define the importance weights by $w_t^{*(i)} = \frac{p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})} = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}^{(i)})p(\mathbf{x}_{0:t}^{(i)})}{p(\mathbf{y}_{1:t})q(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})}$, whose expression requires the calculation of the integral ($p(\mathbf{y}_{1:t}) = \int_{\mathbf{x}_{0:t}} p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})d\mathbf{x}_{0:t}$), which is generally impossible. We can nevertheless show that the following equation is usable [DOU 01]:

$$\begin{aligned}\hat{\mathbb{E}}_p[\mathcal{F}(\mathbf{x}_{0:t})] &= \frac{1}{N} \sum_{i=1}^{N} \mathcal{F}(\mathbf{x}_{0:t}^{(i)}) \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}} \quad \text{with} \\ w_t^{(i)} &\propto \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}^{(i)})p(\mathbf{x}_{0:t}^{(i)})}{q(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})} \qquad [1.7]\end{aligned}$$

This estimator almost certainly converges when N tends to infinity. Then, it is sufficient to make the importance sampling recursive, to obtain the particle filtering algorithm described below.

1.2.2.2. *Particle filter*

The idea is thus to represent and to approximate empirically the *a posteriori* density by a weighted sample of size N $\{\mathbf{x}_{0:t}^{(i)}, w_t^{(i)}\}$, $i = 1, \ldots, N$ such that:

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \approx \frac{1}{N} \sum_{i=1}^{N} w_t^{(i)} \delta_{\mathbf{x}_{0:t}^{(i)}}(\mathbf{x}_{0:t}) \qquad [1.8]$$

where the individuals $\mathbf{x}_{0:t}^{(i)}$, also called particles, are the realizations of the random variable $\mathbf{x}_{0:t}$ (state of the object) in the state space (δ being the Dirac function). Every particle is therefore a possible solution of the state to approximate and its associated weight represents its quality according to the available observations. Hence, the sample $\mathcal{S}_t = \{\mathbf{x}_{0:t}^{(i)}, w_t^{(i)}\}_{i=1}^{N}$ at the instant t is calculated from the previous sample $\mathcal{S}_{t-1} = \{\mathbf{x}_{0:t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N}$, so as to obtain an approximation (via sampling) of the filtering density $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$ at the current instant. For this, three stages are necessary: i) a state exploration stage, during which we propagate the particles via the proposal function, ii) a stage for the evaluation (or the correction) of the particle quality, which aims to calculate their new weight and finally iii) an optional stage for particle selection (re-sampling). The generic particle filtering scheme (SIR filter – *sequential importance resampling*), between the instants $t - 1$ and t, is summarized in the algorithm below.

1) Representation of the filtering density $p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1})$ with a set of particles $\{\mathbf{x}_{0:t-1}^{(i)}, w_{t-1}^{(i)}\}$, $i = 1, \ldots, N$.

2) Propagation, or exploration of the state space, with an importance (or proposal) function:

$$\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t}) \qquad [1.9]$$

3) Correction, or evaluation of the particle quality, with observations, by calculating the weights:

$$\begin{aligned} w_t^{(i)} &= \frac{p(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})} \\ &\propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})p(\mathbf{x}_{0:t-1}^{(i)}|\mathbf{y}_{1:t-1})}{q(\mathbf{x}_{0:t}^{(i)}|\mathbf{y}_{1:t})} \end{aligned} \quad [1.10]$$

Assuming that $q(\mathbf{x}_t^{(i)}|\mathbf{y}_{1:t}) = q(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})q(\mathbf{x}_{0:t-1}^{(i)}|\mathbf{y}_{1:t-1})$, we have:

$$\begin{aligned} w_t^{(i)} &\propto \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})p(\mathbf{x}_{0:t-1}^{(i)}|\mathbf{y}_{1:t-1})}{q(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})q(\mathbf{x}_{0:t-1}^{(i)}|\mathbf{y}_{1:t-1})} \\ &\propto w_{t-1}^{(i)} \frac{p(\mathbf{y}_t|\mathbf{x}_t^{(i)})p(\mathbf{x}_t^{(i)}|\mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)}|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})} \end{aligned} \quad [1.11]$$

The weights are then normalized $\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^N w_t^{(j)}}$.

4) Approximation of the filtering distribution expectancy, the *a posteriori* law at instant t:

$$\mathbb{E}(\mathcal{F}(\mathbf{x}_{0:t})) \approx \frac{1}{N}\sum_{i=1}^{N} \tilde{w}_t^{(i)} \mathcal{F}(\mathbf{x}_{0:t}^{(i)})$$

5) Resampling (if necessary).

The equations below allow us to approximate the trajectory of the objects, but they can also allow to approximate only their state at instant t, by simply integrating over $\mathbf{x}_{0:t-1}$. In practice, this amounts to replacing $\mathbf{x}_{0:t}$ and $\mathbf{x}_{0:t-1}$, respectively, by $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ in the algorithms. In the rest of this work, depending on the applications, either one or the other possibility will be studied.

Once the theoretical framework is defined, we will discuss the problem of visual tracking by particle filtering in the next section.

1.2.3. *Application to visual tracking*

The PF has been used in numerous disciplines, such as communication, networks, biology, economy, geoscience, social sciences, etc. In image processing, it has been used in many domains (medical imagery, video analysis, meteorological imagery, robotics, etc.), for various applications such as segmentations or tracking in video sequences, which is the primary subject of our research.

Visual tracking poses many problems, among which the changes in appearance or illumination, occlusion, the appearance and the disappearance of objects, environmental noise and erratic movements are just a few examples. Particle filtering allows us to represent the arbitrary densities, focusing on specific regions of the state space and managing multiple models. It is easy to implement, robust to noise and to occlusions, although this requires taking a certain amount of precautions, among which:

– the choice of the state model $\mathbf{x}_t$, defined by a set of information that characterizes the object to track;

– the choice of observations $\mathbf{y}_t$, which allow identifying the object to track;

– the definition of an importance (or proposal) function q to propagate particles in a way that will guide the search in the state space;

– the definition of a likelihood function $p(\mathbf{y}_t|\mathbf{x}_t)$, which will link the current state of the object to the observation;

– the choice of the resampling method in order to avoid the problem of degeneration, which we will explain further in this section.

We will later give several solutions suggested by the literature to each of these points.

1.2.3.1. *State model*

The choice of a model x_t for the state depends on the available knowledge and the characteristic differences of the object that we would like to track. In this part, we describe how to model the state x_t of an object.

The most common method to represent an object is to use its geometric characteristics, in particular its position in the image (this is the case of the illustration in Figure 1.3). The $2D$ form can be given by a set of arbitrary points [ARN 05a, ARN 07, VER 05b] or specific points, such as edges [DOR 10, DU 05], contour points [CAR 10, CHE 01, LAK 08, MOR 08, XIA 08] or reference points [TAM 06]. Classical forms are also used, such as rectangles [BRA 07a, HAN 05b, HU 08, LEI 06, LEI 08, PÉR 02, WAN 09] or ellipses [ANG 08, MAG 09, NUM 03a], as well as forms interpolated by splines [LAM 09, LI 04a, LI 03]. We can also use level-sets [AVE 09, RAT 07a] or active contours [RAT 05, RAT 07b, SHE 06]. Finally, more evolved models integrating the relations between sets of pixels [HOE 10, HOE 06] are sometimes used. Among $3D$ forms, we use simple shapes (parallelepipeds, spheres) [GOY 10, MIN 10, MUÑ 10, ROU 10], thin $3D$ mesh of the face [DAI 04, DOR 05], the human body [GAL 06] or the hand [BRA 07c, CHA 08], as well as the contours [PEC 06].

Recently, numerous studies were conducted on the tracking of articulated objects, in which an object was modeled by a set of $2D$ or $3D$ shapes linked between themselves by articulations [BER 06, BRU 07, QU 07, SIG 04, YU 09]. The appearance models are also used, which require learning color [MAR 11, WAN 07], thumbnails [BHA 09], illumination [BAR 09, SMA 07], the exposure [WAN 05] or multiple shapes [BRA 05, GIE 02]. We also find more exotic appearance models using blur [SMA 08] or laser [GID 08] information. Finally, the state can be described by movement information, given by the refined transformations [GEL 04, KWO 08, MEI 09], velocity and/or acceleration [BLA 99a, CUI 07, VER 05a, DAR 08b] (we sometimes talk about auto-regressive models) or the trajectory [BLA 98a].

Naturally, these models are often combined to improve the description of the object, which increases the size of the state space, often making calculations unacceptable. We then need to make a compromise between the quality of the description and the computation time. Figure 1.1 gives several examples of state models used in tracking by PF.

1.2.3.2. *Observation model*

Here, again, the choice of the observation model y_t depends on the available information. In visual tracking, this information is extracted from the images, which are generated by different types of sensors, the number of which can vary. Many approaches work directly on pixels, which are often filtered during a simple pre-processing stage [BHA 09, GEL 04, GON 07, KAZ 09, KHA 06, SCH 07] or simply on pixels of the area from the extracted foreground [CHE 03a]. The difference between these approaches depends on the form of acquisition, which can supply, for example, fluorescent [LAM 09], $2D$ [SHE 06, SMA 08] or $3D$ [CHE 08] microscopic, infrared [PÉT 09] or even ultrasound [SOF 10] imagery. Note that for the color, we primarily use RGB representations [CZY 07, HU 08, MAG 05a, MAG 07, MAR 11, NUM 03a] and HSV [LIU 09, MUN 08b, PÉR 02, PER 08, SNO 09] (the latter being generally more adapted to vision problems, as it is less sensitive to changes in illumination). Other types of sensors are sometimes used, providing information such as distance and depth maps [ARN 05a, BER 06, LAN 06, MUN 08b, ZHU 10], movement maps [SCH 06], laser data [CUI 07, GID 08, GOY 10], projective images [ERC 07], occupation [MUÑ 09] or sound [CHE 03a, PÉR 04] maps. Figure 1.2 gives several examples of these.

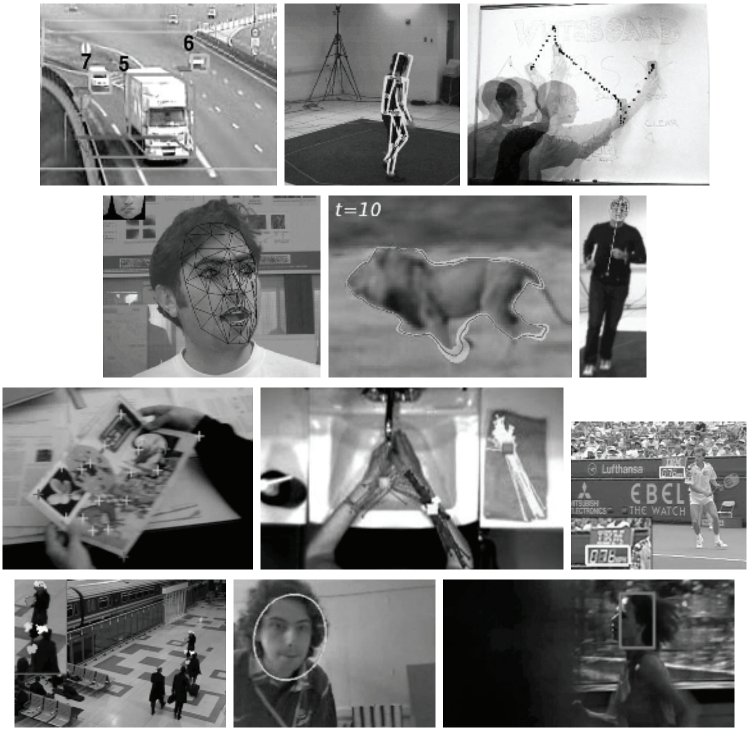


Figure 1.1. *Some examples of state models used to represent the object to track. Form left to right, top to bottom, a model integrating illumination [BAR 09], an articulated model [SIG 10a], a trajectory [BLA 98a], a* $3D$ *facial mesh [DOR 05], level sets [AVE 09], a sphere [ROU 10], a set of points-of-interest [ARN 07], areas and their relations [HOE 10], a rectangle [BRA 07a], edges [DOR 10], an ellipsis [MAG 09] and appearance models [MAR 11]. For a color version of the figure, see www.iste.co.uk/dubuisson/tracking.zip*

1.2.3.3. *Importance function*

The importance function, or the proposal function, $q(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(i)}, \mathbf{y}_{1:t})$ makes it possible to guide particles, between two instances, in the state space in order to have *a priori* approximation of the density of the tracked object. Its choice is essential, since if the particles are propagated in inappropriate areas, the tracking will fail.

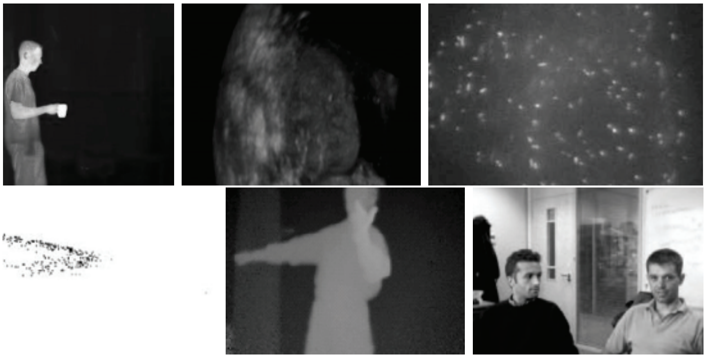


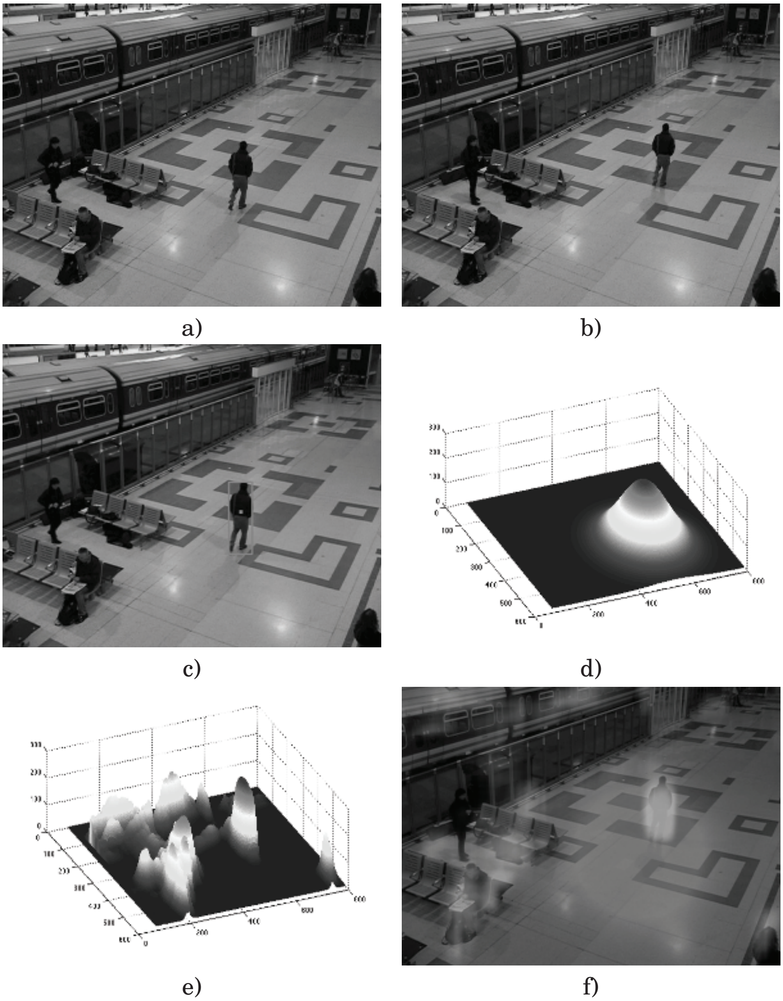
Figure 1.2. *Several examples of images modalities that are used as observations for tracking by PF. From left to right: infrared [PÉT 09], ultrasound [SOF 10],* $2D$ *microscopy [SMA 08], occupation map [MUÑ 09], depth map [ZHU 10] and Charge-Coupled Device (CCD) [PÉR 04]. For a color version of the figure, see www.iste.co.uk/dubuisson/tracking.zip*

The most common choice for the importance function is the transition function $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. In this case, the importance weight is proportional to the likelihood function, as equation [1.11] becomes $w_t^{(i)} \propto w_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$. This filter is more commonly called *Bootstrap* or the CONDENSATION algorithm [GOR 93, ISA 98a]. Here, we do not use the *a priori* information on the changes between the two instants, which are often modeled by two random Gaussian steps around the previous state approximation (see example illustrated in Figures 1.3(d) and 1.3(f)) or auto-regressive models from the first to the second order, integrating information on kinematics (velocity, acceleration). We can also retrieve the transition from the past [ISA 98a]. The problem with sampling the transition function is that if the transition model is inappropriate, most particles generated via this model will be “lost”, as they will not be corrected properly.

Figure 1.3(f) shows the areas of particle diffusion (in blue) and of observation characterization (in red) (see color version of the figure): if these areas do not overlap, then the tracked object will be lost. The necessity to use an optimal importance function for visual tracking becomes obvious once the object can perform sudden movements that we can neither anticipate not model.

Hence, numerous works attempted to best approximate the optimal importance function, which generates particles randomly, and which thus plays an essential role in particle filtering [DOU 01, PIT 99, MER 00]. It has been shown that the optimal importance function (in the sense of minimizing the variance of the sample) needs to integrate the last observation and that it is then written $q(\mathrm{x}_t|\mathbf{y}_t, \mathrm{x}_{t-1})$ [DOU 00b]. Unfortunately, in most computer visualization problems, this expression is unknown and we have therefore searched for other solutions using the current observation. For example, a mix between the classical transition density and the detection function defined by a learning algorithm is used in [LU 09]. Other proposal functions are deducted from learning, for example movement over time [SHO 11]. A simulation taking advantage of an approximation of a displacement field by optical flow of the scene has also been suggested in [ARN 07], and in the partially linear context, allows extracting the optimal proposal function.

Other works suggested migrating the particles toward regions with high likelihood. This is the case of auxiliary particle filtering [PIT 99], which pre-selects the particles to propagate according to their link with the most recent observation. The proposal function of each of the particles can also be defined by an extended [TAN 96] or unscented [JUL 97] Kalman filter, in order to approximate the optimal proposal function by a Gaussian probability density. In the case of likelihood sampling [TOR 04], the particles are sampled directly from the likelihood density.



a) b)

c) d)

e) f)

Figure 1.3. *A simple problem for following the center of a region containing a person between the instants a) $t-1$ and b) t: the state space is defined by the Cartesian position parameters (x, y) of the center of the region (so we have $|\mathcal{X}| = 2$). c) At the instant $t-1$, we have an estimate of the position of the enclosing box (in red) and we wish to estimate its new position at the instant t. d) The proposal function is a random Gaussian walk around the previous approximated position of the object and is used to spread the particles in the state space. e) The likelihood, obtained from the Bhattacharyya distance [BHA 43] between the color distribution in the enclosing box approximated at $t-1$ (image (c)) and in the target histograms, will be able to affect the weight of the particles. f) The influence areas of these two densities in the image space with the proposal function in blue and the likelihood function in red: the higher the tint of the pixel, the more value is attributed to the density. For a color version of the figure, see www.iste.co.uk/dubuisson/tracking.zip*

In fine, there are many proposal functions whose definition is based on the characteristics of the image and that are often related to the problem or the context. Among these *ad hoc* solutions, we can mention the works suggested in [ISA 98b], where an auxiliary contour is used to generate samples, or those described in [PÉR 04], which detect salient points according to precise characteristics (color, movement) that are centered on normal distributions, thus modeling the proposal function by a Gaussian mixture. Finally, in [SUL 01], the particles are guided with the density supplied by the appearance model.

1.2.3.4. *Likelihood function*

The likelihood function gives us a reason to believe in the validity of the observation $\mathbf{y}_t$, given the state $\mathbf{x}_t$ of the object. The way that we perceive this belief leads to first represent synthetically the available information based on the current observation $\mathbf{y}_t$, and then calculate the difference by comparison to the synthetic representation of an ideal situation. For example, this difference $\mathcal{E}$ can measure the similarity or the distance between the model of the previously estimated state and the model of the target corresponding to a hypothesis (or particle). Particle filtering determines the probabilistic framework in which this belief is modeled by a probability density $p(\mathbf{y}_t|\mathbf{x}_t)$, and therefore a common definition of the likelihood function is given by $p(\mathbf{y}_t|\mathbf{x}_t) \propto e^{-\lambda\mathcal{E}^2}$, where λ is the deviation parameter that makes the likelihood more or less pronounced.

There are several detailed studies on likelihood, generally made in a specific context. For example, a study on the choice of measure for similarity to compare color models using histograms is provided in [DUN 10], while the influence of the deviation parameter λ is studied in [LIC 04]. The choice of the likelihood function is directly related to the information used to characterize the tracked object. The literature is abundant on this subject. We often use color models described by histograms, as laid out in the early articles [PÉR 02], as well as many others [JAW 06, NUM 03b, SAT 04, WU 08b], where the Bhattacharyya distance allows measuring the similarity of two histograms (see the illustrations in Figures 1.3(e) and 1.3(f)). Other measurements have also been used, such as the Jensen–Shannon divergence [PER 08], the diffusion distance [LU 09] and the Earth Mover's Distance (EMD) distance [KAR 11]. Likelihood is also based on other types of information, such as the contours [MAC 99a], the foreground [LEI 08], statistics on pixel [LAM 09, MAG 05b, PAL 08], the shape [CAR 10, RAT 07b], the texture [LAO 09], the appearance (combination of shape and texture) [HOE 10, LEI 06, ZHO 04] as well as characteristic features [ARN 05b, DU 05].

Naturally, with the amount of information, the current trend is to combine likelihoods, often by multiplying them, under the independence hypothesis, in order to be able to handle multi-object, multi-characteristic, multi-modal or multi-view tracking problems. Examples of these will be given in Chapter 2.

1.2.3.5. *Resampling methods*

The particle filter and its variants all use a resampling stage in order to avoid the issue of degeneration in particles, that is the cases where the weight of every particle except one is close to zero. In practice, the variance of the importance weight $w_t^{(i)}$ increases over time, which has dramatic consequences on the tracking. There are several resampling methods and their goal is always the same: duplicate particles with high weight and, implicitly, eliminate those with low weight. A theoretical comparison of their advantages and downsides is given in [DOU 05]. Here we outline five methods most used in tracking.

– Multinomial resampling [GOR 93] (the most used) consists of selecting N numbers k_i, $i = 1, \ldots, N$, according to a uniform distribution $\mathcal{U}(0, 1)$. The sample $\mathcal{S}_t = \{\mathbf{x}_t^{(i)}, w_t^{(i)}\}$

is replaced by a new sample $\mathcal{S}'_t = \{\mathbf{x}_t^{(D(k_i))}, \frac{1}{N}\}$ where $D(k_i)$ is a unique integer j such that $\sum_{h=1}^{j-1} w_t^{(h)} < k_i \leq \sum_{h=1}^{j} w_t^{(h)}$. If $(n_1, \dots, n_N)$ indicates the number of times when particles of $\mathcal{S}_t$ are duplicated, then $(n_1, \dots, n_N)$ is distributed according to the multinomial law $\mathcal{M}(N; w_t^{(1)}, \dots, w_t^{(N)})$ (the descendants of the particles conjointly follow a multinomial law). In other words, by sampling with replacement N times the probability $\mathcal{M}\left(1; w_t^{(1)}, \dots, w_t^{(N)}\right)$, we obtain N new particles i.i.d. according to $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, with a weight of $1/N$.

– Stratified resampling differs from multinomial resampling by randomly selecting k_i according to the uniform distribution $\mathcal{U}(\frac{i-1}{N}, \frac{i}{N})$.

– Systematic resampling [KIT 96] randomly selects a number k according to $\mathcal{U}(0, \frac{1}{N})$ and then defines k_i such that $k_i = \frac{i-1}{N} + k$.

– Residual resampling [LIU 98] is very efficient for reducing the variance of the set of particles inducted by the sampling state. First, for each $i \in \{1, \dots, N\}$, $n'_i = \lfloor N w_t^{(i)} \rfloor$ duplicate particles $\mathbf{x}_t^{(i)}$ of $\mathcal{S}_t$ are inserted into $\mathcal{S}'_t$. The remaining $N - \sum_{i=1}^{N} n'_i$ particles are selected randomly according to the distribution $\mathcal{M}(N - \sum_{i=1}^{N} n'_i; N w_t^{(1)} - n'_1, \dots, N w_t^{(N)} - n'_N)$, for example through multinomial resampling. The weights of the $\mathcal{S}'_t$ particles are all equal to $1/N$.

– Weighted sampling is defined as follows. Let $g : \mathcal{X} \mapsto \mathbb{R}$ be a strictly positive continuous function and ρ_t be the multinomial distribution defined by $\rho_t^{(i)} = g(\mathbf{x}_t^{(i)}) / \sum_{j=1}^{N} g(\mathbf{x}_t^{(j)})$ for $i = 1, \dots, N$. Take $k_1, \dots, k_N$, independently according to the probability ρ_t. We construct the set of particles $\mathcal{S}'_t = \{\mathbf{x}_t^{(k_i)}, w_t^{(k_i)} / \rho_t^{(k_i)}\}_{i=1}^{N}$. It has been shown in [MAC 00a] that $\mathcal{S}'_t$ represents the same probability distribution as $\mathcal{S}_t$, focused on the modes of g (the usual choice for g is therefore the likelihood function). Note that, in contrast to other methods, weighted resampling does not affect particles with a weight of $1/N$.

A problem may occur if the sample is resampled too often: the representation decays (*sample impoverishment*), as the highest importance weights are duplicated too many times and the sample is reduced to a single particle. Moreover, every resampling stage diminishes the statistical independence of the particles, which is a strong assumption necessary for the convergence of the filter. Therefore, the decision to resample must be made at an opportune moment, in order to avoid increasing the variance of the sample, as well as to maintain a reasonable number of "good" particles (i.e. with relatively high weights) over time. One solution is to resample when $N_t^{\text{eff}} = \left(\sum_{i=1}^{N}(w_t^{(i)})^2\right)^{-1}$ reaches a threshold value, that is when the amount of "good" particles becomes too small (in practice, often fixed at 75% of N).

1.3. Limits and challenges

There are many versions of the PF [CHE 03b, DOU 01, MAS 01] and its primary appeal is its capacity to process and represent arbitrary densities, maintain multiple hypothesis, take into account non-Gaussian noise and to focus on areas of state spaces. Furthermore, it is relatively simple to implement and extend, robust to "noisy" backgrounds and to occlusions, rendering it rather suitable for problems in vision. In spite of this, the PF suffers from a certain number of downfalls. One of the previously discussed issues is that of particle degeneration, i.e. the loss of particle diversity in a sample, which can only be solved through regular resampling, by using an optimal proposal function or, failing that, a function approaching the optimum, whose choice is therefore critical. The choice of the resampling frequency must also be made with caution. Finally, the optimal number N of particles is impossible to define, as it is mostly dependent of the targeted application.

Nevertheless, the major problem, which remains a large constraints in this methodology, is quite certainly the calculation complexity. Indeed, the amount of particles necessary for good tracking increases exponentially with the size of the spaces in which the state and observation models are defined [MAC 00b]. The first applications of particle filtering in the domain of our interest concerned tracking the position of objects corresponding to the center of an enclosing box [NUM 02, PÉR 02, RUI 01] or contours [MAC 99a, TOR 01], by considering sometimes complex schemes integrating several objects, occlusions or changes of appearance, as shown in Figure 1.4.

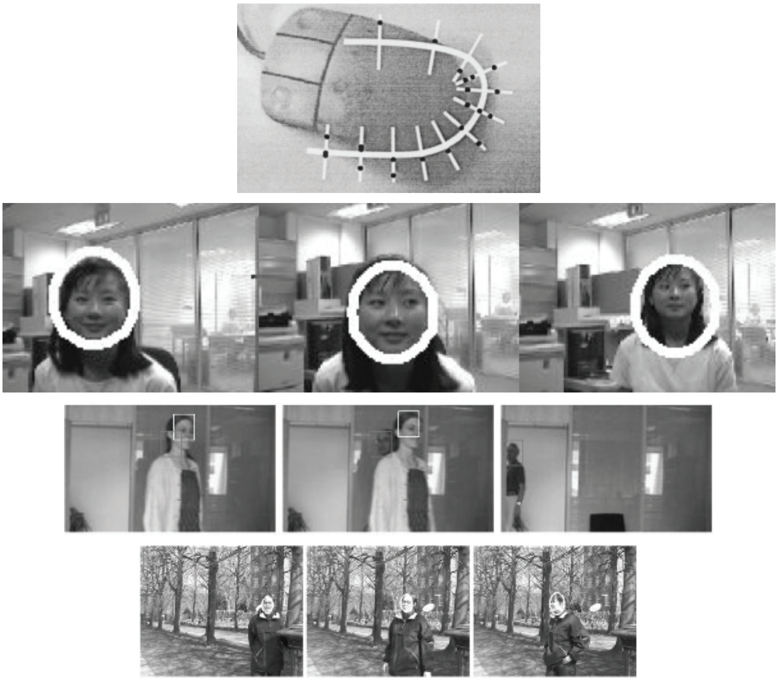


Figure 1.4. *Some results of the tracking obtained by the first suggested approaches, from left to right, top to bottom: tracking an object represented by its deformable contour [MAC 99a], tracking a face with changes in scale [RUI 01], tracking two faces with occlusion management [PÉR 02] and tracking a face with appearance changes [NUM 02]. For a color version of the figure, see www.iste.co.uk/dubuisson/tracking.zip*

The current tendency is to use as much information as possible for tracking, either by mixing different observation sources (several views, several forms of acquisition) or by refining the models that we use for the objects tracked, which requires increasing the number of parameters of the state vector and therefore the size of the space in which the latter is defined. To give an example, tracking human beings requires more and more precision and the models tend toward complex articulated objects similar to those in Figure 1.5(a), which entail a very large state space. The new representations of human movement [GUE 12], as well as the basis for testing tracking algorithms, such as HumanEva [SIG 10a], direct research toward characterization of posture, behavior recognition, gait analysis or even the detection of events such as falling over, which require more and more precise models. We also seek to have a fine analysis of movement and deformation, as illustrated in Figure 1.5(b) for the case of a deformable surface or interacting hands. Finally, taking into account a large number of objects in the same tracking scheme, in particular by integrating relations between them to improve their tracking, is also important. In Figure 1.5(c), we can see two applications for which we need to track a large number of objects simultaneously: for analyzing automatically the game tactics in a sport or for characterizing the behavior of a crowd. Currently, the PF does not allow doing this: when there are too many objects, they need to be tracked individually and a module for measuring interactions must be added as a post-process to make analysis. If a lot of progress aimed toward integrating increasingly complex models (see examples in Figure 1.1) was made over the past 10 years, tracking is sometimes slow and the solutions are often constrained to optimize their implementation (for example simplifyed hypotheses are used to bypass the complexity of certain calculation or models that can only be used for a certain type of application).

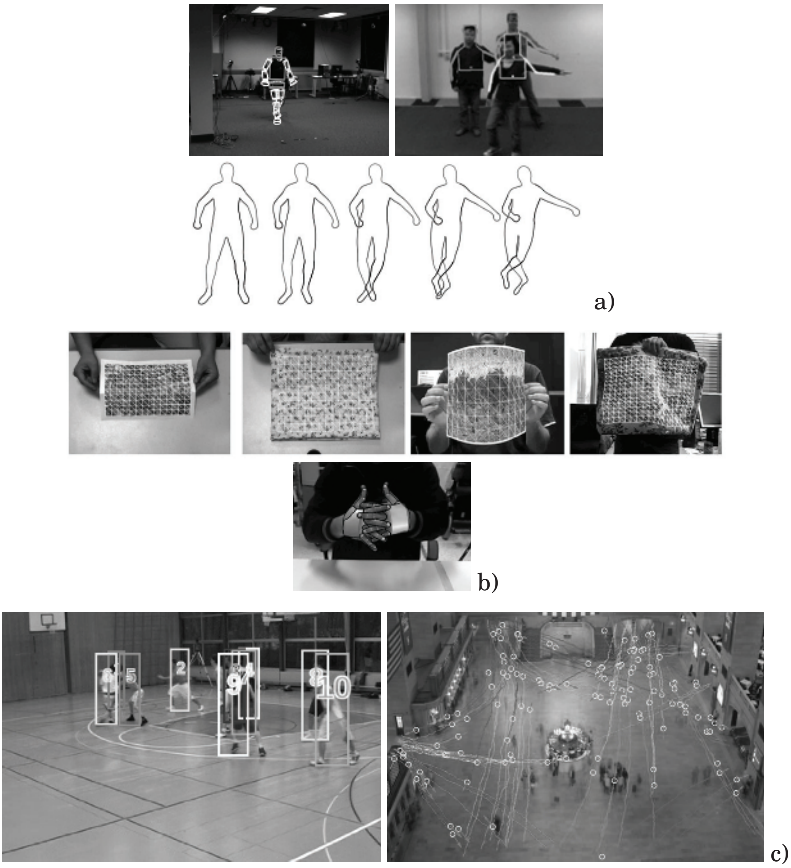


Figure 1.5. *Examples of complex tracking cases that still do not have an efficient solution in particle filtering. a) Complex modeling of a human body, from left to right by articulated object with a fine representation [SIG 10b], by multiple skeletons [BER 11] or by free and deformable shapes [FRE 10]. b) Tracking a deformable* $3D$ *surface [WAN 11a] and modeling the interaction between two hands [OIK 12]. c) Tracking multiple objects in a dense environment, from left to right are cases studied in [HUO 12] and [ZHO 12]. For a color version of the figure, see www.iste.co.uk/dubuisson/tracking.zip*

1.4. Scientific position

The application examples mentioned previously are currently infeasible with particle filtering, as the modeling still requires too many parameters. We could stop at the results provided by other approaches in the domain; however, we believe that new models of PFs that manage large state and observation spaces better would offer new perspectives on the research in the domain of tracking. Indeed, the management of multiple hypothesis naturally integrated into this filter could be fully exploited, if we were not constrained to using representations defined in small state and observation spaces. The management of large state and observation spaces is hence, in our opinion, a major current challenge to tackling future research problems. For this purpose, several avenues can be used. First, the definition of new data representation models, as complete and as compact as possible, is essential. In other words, integrating the wealth of the representation resulting from multiple characteristics used for representing the object in particle filtering is the first avenue to follow. Second, as we stated, particles are spread in the state space with the aim to optimally sample the filter density. Another way to solve the problem of the very large size of the state and observation space is therefore to provide models that more cleverly cover these spaces. For this purpose, we can either focus the search in the state space, in order to only cover interesting areas, or decompose it in sub-spaces, where sub-calculations can be made. We provide solutions using these different avenues in Chapters 2, 3 and 4 of this book.

1.5. Managing large sizes in particle filtering

The algorithms that tackled the problem of large sizes of state spaces can be roughly divided into three primary classes: those that reduce the size of the space, often by adding constraints to the model, those that use a local search and finally those that decompose the space into sub-spaces of a lesser size.

The first class of approaches adds constraints to the mathematical model in order to reduce the size of the state space to explore. In many articles, this is accomplished by introducing the physical constraints of the articulate object into the tracking model [BRU 09, OIK 11]. In particular, this type of approach is very popular in human tracking. Hence, in [VON 08], the constraints are introduced during a simulation stage, while in [BRU 07, HAU 10b], they are included directly in the proposal function constructed specifically to follow a person. Other approaches adding constraints add *a priori* information to the object [HAU 10a, COV 00, HAU 11], exploit the knowledge of its behavior [DAR 08a] or take into account its interaction with the environment [KJE 10].

The second class of approaches, often known as optimization oriented approaches, is a set of algorithms that combine particle filtering with local search techniques [MIN 86]. Given the stochastic nature of the filter and the combinatorial size of the state spaces of the objects, the PF is never guaranteed to produce a set of particle positions sufficiently close to the modes of the density to approximate. Thus, combining the filter with local search techniques can improve significantly its capacity to focus on these modes. For this reason, optimization approaches are very popular within the community working on object tracking. Among these techniques, we can mention the gradient descent methods that were specifically studied in this context. For example, stochastic gradient descent was successfully combined with particle filtering [HOF 11] and new stochastic meta-descent approaches were suggested in a constrained space [BRA 07b], giving the efficient *smart particle filter* [BRA 7d]. Particle swarm optimization techniques are also used conjointly with the filter [WAN 06, LI 11, JOH 10, KRZ 10]. Here, the idea is to apply evolutionary algorithms inspired by social animal behavior (birds, fish, bees, ants, etc.) to evolve the particles in accordance with their own experience, as well as that of their neighbors (for a complete review of these techniques, we recommend the lecture of [DAS 11]). Similarly, the introduction of population-oriented meta-heuristics and genetic algorithms was used in the context of particle filtering [SÁN 05a]. Simulated annealing was also introduced into particle filtering, giving the renowned *annealed PF* (APF) (or particle filtering with simulated annealing) [DEU 05]. APF adds iteration of pseudo-simulated annealing to the resampling, in order to spread the particles in the state space and hence position them closer to the modes of the density to approximate. Naturally, there are other optimization-oriented methods (such as the scatter search [PAN 08a], etc.) that we cannot list here, as they are not the main subject of this work. Nevertheless, all of these approaches share the commonality of being a compromise between the quality of the approximation of the estimated density and the velocity of convergence. Unfortunately, given their local nature, although many of these approaches may converge rapidly to a local minimum near their starting point, they require a lot more time to converge to the overall minimum, which, in addition, is not systematically guaranteed. Hierarchic approaches tried to resolve these problems by adopting a strategy of progressively refining the search space, starting from a general description of the state space to end on a finer one, giving a complete description. The *progressive particle filter* [CHA 10] is an example of this. Finally, it is important to mention that all of these methods assume that every required observation is available at every instant, which is unfortunately not necessarily the case in practice.

The third class of approaches exploits the natural decompositions of state and observation spaces into a set of sub-spaces of a more reasonable size, where particle filtering can be applied. Partitioned sampling (PS) [MAC 99b] is probably the best known here. It uses the fact that in many problems, both the dynamics and the likelihood can be decomposed. The key idea is to substitute the application of the filter to the whole space with a sequence of applications to the sub-spaces, therefore significantly accelerating the process. Despite recent improvements [SMI 04, DUF 09], PS suffers from too high a number of necessary resamplings, which increases the noise, leading to a decrease in tracking quality over time. An equivalent type of decomposition is used in [KAN 95], in the context of dynamic Bayesian networks (DBNs). Here, the proposal density of the predictive stage is decomposed as a product of conditional distributions in each node of the DBN at the current instant. The predictive stage is then executed iteratively on each node of the network, according to the topological order by using the proposal distribution of the node based on its parent in the network. In [ROS 08], the sampling idea suggested in [KAN 95] is combined with the resampling scheme from [MAC 99b] in order to create a particle filtering algorithm well adapted to DBN. This algorithm can be seen as a generalization of PS. By following the topological order of the network for sampling and resampling the particles every time a node is explored, the particles with low likelihood in a sub-space are excluded, while those with a high likelihood are multiplied. The effect is similar to that of the weighted resampling in PS. Another approach coming from the Bayesian community is the *non-parametric belief propagation* algorithm [SUD 10, ISA 03]. It combines particle filtering with the *loopy belief propagation* algorithm [PEA 88, YED 05], in order to accelerate calculations (at the price of worse approximations). It was successfully applied to the large-size problem [SUD 04, SIG 03, BER 09, IHL 04, LAN 06]. Another popular approach is the *Rao-Blackwellized PF* (RBPF) *for DBN* [DOU 00a]. By using the natural decomposition of joint probability, RBPF decomposes the state space into two parts, following these conditions: the distribution of the second part, based on the first, can be estimated with a Kalman filter. The distribution of the first part is estimated by particle filtering. As the size of the first part is smaller than the whole space, its sampling stage needs less particles, and the variance of the estimation can therefore decrease. Although RBPF is very efficient at reducing the size of the problem, it cannot be applied to any Bayesian network, as the state space cannot always be decomposed in the manner that it assumes (i.e. assuming a part of the space to be linear). The work suggested in [BES 09] is a parallelized PF dedicated to Bayesian networks, which uses the same probabilistic decomposition of the joint distribution in order to reduce the number of particles required for tracking. The state space is divided into a set of independent sub-spaces. The particles are hence generated independently in the sub-spaces according to different proposal densities. This approach offers real flexibility in terms of proposal density choices. Nevertheless, their definition requires the underlying Bayesian network to have certain structural properties, which limits the generalization of this algorithm.

1.6. Conclusion

We presented in this chapter the elements fundamental to the introduction and the definition of sequential Monte-Carlo methods, as well as their use in the context of tracking in video sequences. In particular, we have shown to which point the community was active in tracking with PF, making it evolve very fast. Nevertheless, this also allowed us to highlight a certain number of challenges that motivated our research. Specifically, the management of large state and observation spaces, which we see as a major challenge to undertake in the years to come, caught our attention. Later in this work, we discuss several solutions that allow us to advance in this direction. These methodological fields affect, on the one hand, the modeling of the data to process, as well as its representation, and, on the other hand, the exploration of the state space by focusing or by decomposition.