



HAL
open science

Enhancement of the state space exploration in sequential Monte Carlo methods for visual tracking

Mehdi Oulad Ameziane

► **To cite this version:**

Mehdi Oulad Ameziane. Enhancement of the state space exploration in sequential Monte Carlo methods for visual tracking. Automatic. Ecole Centrale de Lille, 2017. English. NNT : 2017ECLI0007 . tel-03157858

HAL Id: tel-03157858

<https://theses.hal.science/tel-03157858>

Submitted on 3 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre: 321

CENTRALE LILLE

THÈSE

Présentée en vue
d'obtenir le grade de

DOCTEUR

En

Spécialité : Automatique, Génie informatique, Traitement signal et images

Par

Mehdi OULAD AMEZIANE

DOCTORAT DELIVRE PAR CENTRALE LILLE

Titre de la thèse:

**Enhancement of the state space exploration in sequential Monte Carlo
methods for visual tracking**

Soutenue le 23/06/2017 devant le jury d'examen:

Président	Jean-Charles Noyer, Prof. à l'ULCO
Rapporteur	Séverine Dubuisson, MCF HDR à l'UPMC
Rapporteur	Christophe Ducottet, Prof. à l'Université de Saint-Etienne
Membre	Lyudmila Mihaylova, Prof. à l'Université de Sheffield
Membre	Ariane Herbulot, MCF à l'Université Paul Sabatier
Directeur de thèse	Emmanuel Duflos, Prof. à Centrale Lille
Co-encadrante de thèse	Christelle Garnier, MCF à IMT Lille Douai

Thèse préparée dans le Laboratoire CRISAL

Ecole Doctorale SPI 072 (Lille I, Lille III, Artois, ULCO, UVHC, Centrale Lille)

I would like to dedicate this thesis to my loving parents.

Acknowledgements

First I would like to express my gratitude to the jury members: the President of the jury Pr. Jean-Charles Noyer, the reviewers Pr. Christophe Ducottet and Dr. Séverine Dubuisson, the examiners Dr. Ariane Herbulot and Pr. Lyudmila Mihaylova for their comments and advices which made my thesis works greater.

Also I would like to thank my PhD advisors Pr. Emmanuel Duflos and Dr. Christelle Garnier for their conduct and follow during these three years of my PhD project. I'm grateful for their presence and kindness, their support and their constructive observations. I had the chance to learn from their experience and know how. During this thesis I had the opportunity to share views with them, discuss ideas and grow in both scientific and human levels.

Here also I would like to pay tribute to Pr. Yves Daignon who was my PhD advisor until 2016. His contribution to my work is invaluable, his strength and dedication to science imposes respect. I am honoured to count among his students. May he rest in peace.

Moreover I express my gratitude to the members of the Sigma team for welcoming me in their research group, and for their help their advices and contributions to my work.

I am also grateful to my family and friends for their continuous support and encouraging. Their presence is priceless and without them I would not made it through.

Finally I would like to thank all the people in IMT Lille Douai and in Centrale Lille who made the working environment optimal and friendly, and made this PhD thesis a great experience for me.

Mehdi Oulad Ameziane

Abstract

In this thesis, we focus on the problem of object tracking in image sequences, using Bayesian estimation and specifically sequential Monte Carlo (SMC) methods. SMC methods are currently widely used for visual tracking. These methods have gained a great interest in recent years for object tracking as they have the advantage of handling complex non linear models, they offer a degree of robustness by taking into account the model uncertainties and they are easy to implement.

In this context, we propose to enhance the proposal function, in order to guide the samples in the areas of interest of the state space. The choice of the proposal is essential in SMC methods as their efficiency strongly depends on the exploration of the state space conducted by the proposal.

The simplest proposal is the prior density related to the dynamic model. Due to unpredictable movements in real life scenarios, this prior proposal is not sufficient to explore efficiently the state space. The best choice is the optimal proposal which takes into account the current observations to draw the samples. Unfortunately in visual tracking, computing and sampling from the optimal proposal is impossible because the computational cost is prohibitive.

In order to efficiently explore the state space, our approach is to derive a close approximation of the optimal proposal. The proposed near optimal proposal relies on an approximation of the likelihood which is the most computationally expensive component in the expression of the optimal proposal. We consider a new form of likelihood based on soft detection information which requires less calculations than

the usual likelihoods. Soft detection information reflects probabilities about the object location and is more reliable than the final binary detection. It has already been used in particle filters (PFs) for particle weighting, but according to our knowledge, it has never been exploited for particle drawing.

From the near optimal proposal, we then derive the corresponding tracking algorithms in PF and Markov Chain Monte Carlo (MCMC) frameworks.

Improving the exploration of the state space is most required in two visual tracking applications: abrupt motion tracking and multiple object tracking.

Abrupt motion occurs when the object has unpredictable dynamics with potentially large location variations between successive frames. Thanks to the soft detection information, the proposed near optimal SMC methods are of high interest to capture the motion discontinuity. In this thesis, we focus on their ability to deal with abrupt motion situations and we compare them to the state-of-the-art methods proposed in the visual tracking literature for these situations.

Moreover we extend the near optimal proposal to multiple object tracking and show the benefit of using the near optimal PFs and MCMC algorithms to cleverly explore the high-dimensional state space with a limited number of particles.

Finally we propose to use the local PF recently developed by Rebeschini et al. to further improve the tracking performance. The local PF allows to combine the interaction modelling and the partition of the large state space into separate subspaces of smaller dimension.

These contributions have been evaluated on real and synthetic sequences, and the experimental results show the relevance of our approach in comparison with other methods.

Contents

Contents	v
List of Figures	ix
1 Introduction	1
Nomenclature	1
2 About Visual tracking	8
2.1 Definition and challenges of visual tracking	8
2.2 Classification of visual tracking methods	10
2.2.1 Non rigid representation	11
2.2.2 Rigid representation	12
2.3 Context of the work	15
2.4 Chapter Conclusion	15
3 Sequential Monte Carlo methods	17
3.1 Bayesian estimation	18
3.2 Particle Filtering	19
3.2.1 Importance sampling	21
3.2.2 Sequential importance sampling (SIS)	22
3.2.3 The weight degeneracy problem	23
3.2.4 Resampling	24
3.2.5 Choice of the importance function	27
3.2.6 Auxiliary particle filter (APF)	29
3.2.7 Marginal particle filter (MPF)	31

3.3	Sequential Markov chain Monte Carlo Methods	33
3.3.1	MCMC methods	34
3.3.2	Sequential MCMC	35
3.3.3	Choice of the proposal	39
3.3.4	Adaptive MCMC	40
3.4	Chapter Conclusion	41
4	SMC methods applied to visual tracking	42
4.1	Introduction	42
4.2	Problem formulation	43
4.2.1	State model	43
4.2.2	Dynamic model and prior density	45
4.2.3	Observation model and likelihood	46
4.2.3.1	Appearance models	47
4.2.3.2	Detection information	48
4.3	Selected models for single object tracking	50
4.4	Chapter Conclusion	52
5	Near Optimal Particle Filters	53
5.1	Optimal proposals	54
5.1.1	Optimal importance function for PFs	54
5.1.1.1	Specific cases	55
5.1.1.2	General case	56
5.1.1.3	Suboptimal strategies - state of the art	57
5.1.2	Optimal proposal for sequential MCMC	59
5.1.2.1	Suboptimal strategies - state of the art	60
5.2	Proposed approach : approximation of the optimal proposals	62
5.2.1	Information exploited for sampling	62
5.2.2	Simplifying assumptions	64
5.2.3	Approximations of the distributions of interest	65
5.3	The near optimal particle filter and its variants	66
5.3.1	Near optimal particle filter (NOPF)	66
5.3.2	Near optimal auxiliary particle filter (NOAPF)	69

5.3.3	Near optimal marginal particle filter (NOMPF)	71
5.3.3.1	First algorithm	72
5.3.3.2	Second algorithm	73
5.4	The near optimal sequential MCMC (NOMCMC)	75
5.4.1	First algorithm	75
5.4.2	Second algorithm	78
5.5	Conclusion	80
6	Application to abrupt motion tracking	82
6.1	Performance metrics	84
6.2	Tracking algorithms	85
6.2.1	Basic SMC methods	85
6.2.2	SMC methods to approach the optimal proposal	86
6.2.3	SMC methods to deal with abrupt motion	87
6.2.4	General experimental settings	89
6.3	Performance of near optimal SMC methods	89
6.3.1	Test Sequences	90
6.3.2	Performance of the NOPF	93
6.3.3	Performance of the NOMCMC	97
6.3.4	Comparison of the different near optimal SMC methods . .	100
6.3.4.1	Benefits of particle preselection in NOAPF	102
6.3.4.2	Computational time	103
6.4	Comparison with the state-of-the-art methods for abrupt motion tracking	104
6.4.1	Test sequences	105
6.4.2	Performance comparison	107
6.5	Chapter Conclusion	115
7	Multiple object tracking	116
7.1	Related work	117
7.2	Near optimal proposal for multi-object tracking	118
7.2.1	Problem formulation - Tracking of multiple independent objects	119

7.2.2	Multiple independent near optimal particle filters	120
7.2.3	Joint near optimal particle filter	122
7.2.4	Experimental results	123
7.3	MOT using the local particle filter	127
7.3.1	Problem formulation - Tracking of multiple interacting ob- jects	127
7.3.2	The local particle filter	128
7.3.3	Experimental results	132
7.4	Chapter Conclusion	134
8	Conclusions and perspectives	136
9	Résumé en Français	141
9.1	Contexte de la thèse	141
9.2	État de l'art	143
9.3	La loi de proposition "Near Optimal"	144
9.3.1	Loi de proposition optimale	144
9.3.2	Approximation de la loi de proposition optimale	144
9.3.3	Variantes de l'algorithme NOPF	146
9.4	Application au suivi des mouvements abrupts	147
9.4.1	Comparaison du NOPF avec des PFs standards	147
9.4.2	Comparaison avec des algorithmes de référence	148
9.5	Application au suivi multi-objets	149
9.5.1	Algorithme NOPF pour le suivi multi-objets	149
9.5.2	Local PF pour le suivi multi-objets	150
9.6	Conclusion	152
	Appendix A	153
	References	156

List of Figures

2.1	Taxonomy of tracking methods	10
2.2	Example of non rigid representation : "Tracking a flexing hand across a clutter desk." Illustration from [41].	11
2.3	Examples of point correspondence : "Tracks obtained for a rotating ball and dish sequences." Illustration from [99].	13
2.4	Examples of kernel representations : "Patch based object representation." Illustration from [120].	14
3.1	Dependence graph of a hidden Markov model.	18
3.2	Particular approximation of a density $p(x_t y_{1:t})$	20
5.1	Soft detection likelihood for different size values, (a) ground truth value, (b) 20% larger value and (c) 20% smaller value.	65
5.2	Construction of the near optimal proposal	68
6.1	Image extracted from the "Walking" sequence with the corresponding soft detection map.	90
6.2	Image extracted from the "Corridor" sequence with the corresponding soft detection map.	91
6.3	Image extracted from the "Lemming 1" sequence with the corresponding soft detection map.	92
6.4	Image extracted from the "Lemming 2" sequence with the corresponding soft detection map.	92
6.5	Detection maps of the walking sequence (left to right: original image, soft detection map, hard detection map)	93

LIST OF FIGURES

6.6	Tracking results obtained with PF (green), BPF (red) and NOPF (black) for a DS rate of = 10 (left to right: "Walking", "Corridor", "Lemming 1" and "Lemming 2" sequences).	96
6.7	F-measure per frame obtained by the WLMC and NOMCMC during the "Lemming 2" sequence.	99
6.8	Near optimal proposals obtained with different Gaussian prior densities (different means, same variance).	102
6.9	Running time of the near optimal trackers expressed in frames per second (FPS).	104
6.10	Image extracted from the "Tennis" sequence with the corresponding soft detection map.	106
6.11	Image extracted from the "Ping Pong" sequence with the corresponding soft detection map.	106
6.12	Image extracted from the "Youngki" sequence with the corresponding soft detection map.	107
6.13	Tracking results obtained with NOPF (black), NOMCMC (green), NOMPF (blue) and NOAPF(red) in the "Tennis" sequence. . . .	111
6.14	Tracking results obtained with NOPF (black), NOMCMC (green), NOMPF (blue) and NOAPF(red) in the "Ping Pong" sequence. . .	112
6.15	Tracking results obtained with NOPF (black), NOMCMC (green), NOMPF (blue) and NOAPF(red) in the "Youngki" sequence. . .	114
6.16	Soft detection maps obtained on the frames #118 and #397 of the "Youngki" sequence.	115
7.1	Image extracted from the BIWI sequence and the corresponding soft detection map.	124
7.2	Image extracted from the ILAB sequence and the corresponding soft detection map.	124
7.3	Dependence graph of a HMM satisfying the factorization (7.14). . .	129
9.1	Graphe des dépendances dans un modèle de Markov caché.	142
9.2	Vraisemblance de détection calculée pour différentes tailles de fenêtre: (a) taille réelle, (b) plus grande de 20% et (c) plus petite de 20% .	145
9.3	Success rate pour la séquence YoungKi.	148

Chapter 1

Introduction

One of the most important issues in signal processing is to extract useful information from a set of noisy observation data provided by one or several sensors. This process can be formalised by a mathematical modelling of the observed phenomenon, in which the information of interest is considered as the hidden parameters. The task is then to estimate these unknown parameters from the observations. In this thesis, we focus on the problem of visual tracking, i.e. object tracking in image sequences, using Bayesian estimation and specifically sequential Monte Carlo (SMC) methods.

Sequential Monte Carlo methods

The Bayesian inference permits to estimate the value of one or several parameters, considered as random variables and represented by a state vector. In visual tracking, we are interested in the filtering process which aims to estimate the current state from the current and past observations. The sequential implementation of Bayesian filtering consists in recursively estimating the filtering distribution at each time from two functions: the transition density, which describes the dynamic evolution of the state between two successive times, and the likelihood, which measures the adequateness between the current state and the observations.

When these functions are linear and Gaussian, the estimation problem can be

solved analytically using Kalman filters. Otherwise, in most cases, SMC methods are required to solve the problem. They are based on the approximation of the filtering density with a finite set of samples. These methods are divided into two main families: the importance sampling (IS) methods, also called particle filters (PFs), approximate the filtering distribution by a set of weighted samples called particles and the Markov chain Monte Carlo (MCMC) methods construct a Markov chain that converges to the filtering density. This density is then approached by a sequence of unweighted samples.

SMC methods are currently widely used for visual tracking. These methods have gained a great interest in recent years for object tracking and have shown capabilities to adapt to various tracking schemes. They have the advantage that they can handle complex non linear models, they offer a degree of robustness by taking into account the model uncertainties and they are easy to implement.

Exploration of the state space

In practice, the efficiency of SMC methods strongly depends on the proposal density used to explore the state space, thus the choice of the proposal is essential. Indeed all SMC methods include a sampling step which aims to propagate samples in the areas of interest of the state space between two times. This step is followed by a weighting stage in PFs and by an accept/reject stage in MCMC methods. In PFs, with a well chosen proposal, high importance weights are assigned to most particles, which avoids the problem of weight degeneracy. In MCMC context, the proposal affects the convergence speed of the algorithms. A good proposal increases the acceptance probability of the candidate samples and decreases the mixing time of the generated Markov chain.

The simplest and most commonly used proposal is the transition density, also called the prior density, because it assumes a priori knowledge on the state dynamic evolution. The dynamic models considered in the literature are either very simple, including Gaussian random walks and autoregressive models, or very specific, designed for a particular application. If the model is imprecise and does not reflect the real movement of the tracked objects, most of the samples generated from the prior density are propagated in unlikely areas of the state space

and are wasteful. In PFs, the importance weights associated to most particles are close to zero and in MCMC approaches, most candidate samples are rejected causing the Markov chain to stand still most of the time.

The best choice is the optimal proposal [22] which takes into account the current observations to draw the samples. Unfortunately in most applications, computing and sampling from the optimal proposal is impossible because the analytic form is unavailable or the computational cost is prohibitive. The challenge is then to select the most relevant observations and to make the best use of them to efficiently propagate the samples in the state space.

In visual tracking, the raw data, which include all the pixel values of the sequence images, are too large and too complex to be directly used as the observations. An image pre-processing is needed to extract a more succinct information. Generally, the observations are based on appearance models (colour, texture, gradient, ...) or provided by detection algorithms. The latter are the most widely used for state space exploration. Detection information is provided as silhouette shapes [72, 63, 127] or specific locations [80, 105], such as salient points, which are used to guide the particles in likely regions of the images. But whatever the method used by the detector (image segmentation, background subtraction, supervised learning, ...), the result is not completely reliable and can yield false or missed detections.

In this thesis, we propose to enrich the observation model with soft detection information. This intermediate information, obtained in detection algorithms before hard decision (classification), reflects probabilities about the object location. It is more reliable than the final binary output which rules on the presence or absence of the object of interest. Soft detection information has already been used in PFs for particle weighting [7], but according to our knowledge, it has never been exploited for particle drawing and state space exploration.

In the literature, several suboptimal strategies have been proposed to use the observations in the sampling step and approach the optimal proposal. They can be classified into two main categories. Implicit approaches use the prior density as the proposal and add a step of preselection [81], MCMC [31, 33] or

optimization [65, 20, 45] to guide the particles in the most likely areas of the state space from the observations. The drawback is that some methods leave the theoretical framework of SMC methods, since each additional particle move can potentially alter the filtering density. Explicit approaches are more direct and aim to build or to adapt the proposal from the observations. The limitation is that the proposal is restricted to a Gaussian model [87, 91] or a Gaussian mixture model between the prior density and densities centred on specific points obtained by detection [72, 63, 127, 80, 105].

In the thesis, our approach to efficiently explore the state space is even more direct and aims to derive a close approximation of the optimal proposal. The proposed near optimal proposal relies on an approximation of the likelihood which is the most computationally expensive component in the expression of the optimal proposal. We consider a new form of likelihood based on soft detection information which requires less calculations than the usual likelihood defined from a distance between appearance models. In comparison with previous works, the proposal is no longer limited to the Gaussian model and offers a good compromise between computational complexity and optimality. From the near optimal proposal, we then derive the corresponding tracking algorithms in PF and MCMC frameworks.

Abrupt motion tracking and multiple object tracking

Improving the exploration of the state space is most required in two visual tracking applications: abrupt motion tracking and multiple object tracking. Abrupt motion refers to situations where the object location is subject to unpredictable and large variations between two successive frames. This arises when the object has a fast motion, unexpected dynamics or sudden dynamic changes. Abrupt motion is also due to camera switching or low frame rate, as in many video surveillance applications using IP cameras. In this case, the motion smoothness assumption does not hold anymore and the portion of the state space which needs

to be explored in the sampling step is a priori very large.

The proposed near optimal PF and MCMC algorithms are of high interest to handle abrupt motion. By taking into account the soft detection observations, they are able to capture the motion discontinuity and to guide the samples in the most likely areas of the state space. In the thesis, we focus on the ability of the near optimal SMC methods to deal with abrupt motion situations and we compare them to the state-of-the-art methods proposed in the visual tracking literature for these situations.

Multiple object tracking (MOT) is still a challenging task in visual tracking. The problem of MOT can be solved in two different ways. A common assumption is that each object moves independently of the others. In this case, it is possible to run multiple independent PFs or MCMC algorithms, one for each object. In comparison with single object tracking, the dimension of the state space remains unchanged and tracking performance is similar. However in practice, this assumption is not always correct and this approach is susceptible to tracking failures when the objects interact. Using an interaction model requires to consider a joint configuration space and to run a single joint PF or MCMC algorithm. In this second approach, when the number of tracked objects increases, the state space becomes increasingly large, the number of particles required to find the areas of interest grows exponentially with the state space dimension and the computational cost quickly becomes too expensive.

In the thesis, we extend the near optimal proposal to MOT scenarios and show the benefit of using the near optimal PFs and MCMC algorithms for MOT. By taking into account the soft detection information, they are able to concentrate the particles in the regions of interest of the state space and thus to cleverly explore the high-dimensional state space with a limited number of particles. To improve the tracking performance further while exploiting the dependencies between the objects, we also propose to use the local PF recently developed by Rebeschini et al. [84]. The initial idea is that interactions are local: the dynamics and observations related to an object depends only on the neighbouring objects. The local PF allows to combine the interaction modelling and the partition of the large state space into separate subspaces of smaller dimension.

Sketch of the thesis

This thesis manuscript contains six chapters, including two states of the art and three chapters dedicated to our contributions. Finally a last chapter concludes this work and opens new perspectives. This manuscript is organised as follows.

In Chapter 2, we define the visual tracking task, present a classification of the existing tracking methods and motivate the tracking approach adopted in the thesis.

Chapter 3 deals with Bayesian inference and explains the main SMC methods including PF and sequential MCMC. We discuss the main limitations of these methods and introduce the most commonly used improvements, such as resampling, auxiliary PF, marginal PF and adaptive MCMC. Emphasis is also placed on the crucial importance of the proposal used to explore the state space.

The formulation of the visual tracking problem in the Bayesian framework is addressed in Chapter 4. We review the main models and distributions proposed in the visual tracking literature. They include state, dynamic, observations models and prior, likelihood, proposal densities. Then we present and justify the models we have selected in our tracking scheme, in particular the soft detection based observations and the associated likelihood.

Chapter 5 focuses on the near optimal proposal, which is the main contribution of the thesis. After recalling the notion of optimal proposal in the case of PF and MCMC and reviewing the suboptimal strategies proposed in the literature, we present the assumptions that allow us to derive a close approximation of the optimal proposal. Then, the resulting tracking algorithms in PF and MCMC frameworks are described.

Experimental results realised in the context of abrupt motion are given in Chapter 6. The aim is to show the ability of the proposed near optimal SMC methods to deal with various abrupt motion situations. Our algorithms are compared to the state-of-the-art methods proposed for these situations.

Chapter 7 addresses the MOT problem. We extend the near optimal proposal to MOT scenarios and show the benefit of using the near optimal PFs and MCMC

algorithms. To overcome the high dimension of the problem while exploiting the interactions between the objects, we also propose to use the local PF recently developed by Rebeschini et al. [84].

Chapter 2

About Visual tracking

Visual tracking is one of the most important and fundamental task in computer vision applications such as intelligent surveillance, traffic monitoring, human-computer interaction, vehicle navigation ... In this chapter we first introduce and define the visual tracking task. Then we review some of the major difficulties encountered when performing visual tracking. These challenges are the major drivers of the active research in visual tracking and motivates our work and contributions. Furthermore we present a taxonomy of existing tracking methods, based on two main criteria: the object representation and the tracking mechanism. Finally we motivate our choice concerning the tracking approach considered in this thesis.

2.1 Definition and challenges of visual tracking

In general, visual tracking aims to locate, identify and isolate the image portion representing objects of interest in a sequence of images and construct its trajectory: that is, starting from an initial position and appearance, find the location of the object in all successive frames.

The difficulties in visual tracking come from several factors related to the application context such as:

- Appearance and shape changes due to the temporal evolution of the object.

2. About Visual tracking

These variations can be related to the object itself (person walking for instance) or to the camera perspective.

- Global or local illumination changes. The brightness of the scene can considerably vary due to weather conditions such as sun position, cloud movement, shadowing effect, use of lightening equipments etc.
- Abrupt object motion due to a sudden behaviour change, to camera switching or to low frame rate videos which is the case of IP cameras for instance.
- Partial or total occlusions, which occur when an object overlaps with one another from the perspective of the camera or is hidden by any obstacle in the scene.
- Camera motion or switching, which introduces major changes in the background and foreground of the whole scene and makes the tracking environment unstable.
- Number of objects of interest, in case of multiple object tracking, that can be large. It also evolves over time and needs specific mechanisms to manage the apparition and disappearance of objects.

These are the major, and certainly not all, underlying challenges when performing visual tracking in real-life scenarios. All these issues can be summarised as non-stationarity of the object appearance, dynamics and of the environment. Thus the aim of visual tracking can be formulated as follow: we seek to follow an object from its appearance in a context where this object and its environment are potentially subject to high variations.

Overcoming these problems is the motivation of numerous studies, making visual tracking one of the most active research areas in computer vision and leading to a wide range of solutions and a considerable variety of trackers.

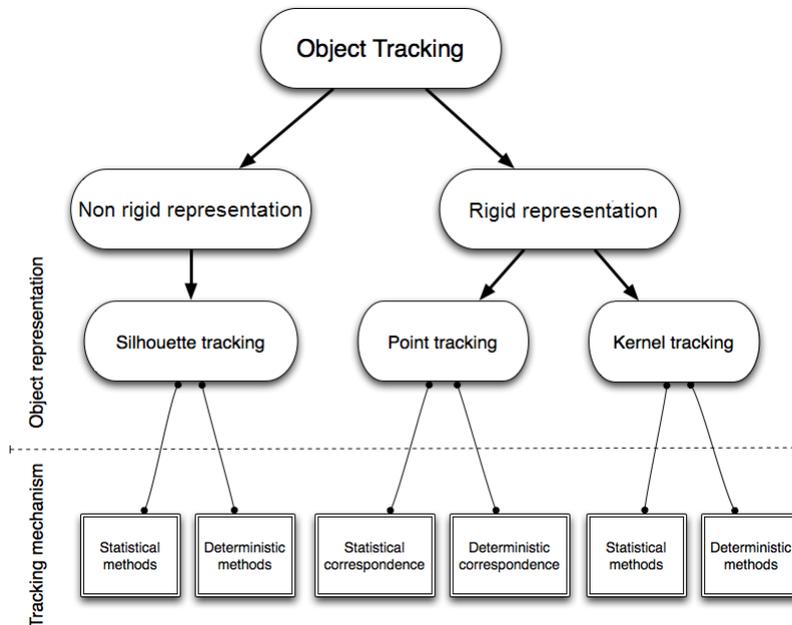


Figure 2.1: Taxonomy of tracking methods

2.2 Classification of visual tracking methods

Several works have been dedicated to survey, compare and benchmark the different contributions made during the past decades [120, 117, 100, 118]. Our classification of tracking methods is inspired from the excellent survey made by Yilmaz and Al.[120].

Visual tracking constitutes a basic task required by higher level computer vision applications. Each application does not expect the same output from the trackers. For example, gesture analysis needs a tracker that records the shape evolution of a person, while intrusion detection or trajectory analysis requires a tracker that outputs only the coordinates of the object of interest. From the diversity of high level applications and their different requirements, we obtain a diversity of tracking methods.

In Figure 2.1 we propose a taxonomy of tracking methods inspired from [120], based on two discriminative criteria, the object representation and the tracking mechanism. In this figure the first branching is based on object representation.



Figure 2.2: Example of non rigid representation : "Tracking a flexing hand across a clutter desk." Illustration from [41].

Non rigid representation takes into account, in the tracking formulation, all the shape complexity of the object of interest and thus it allows to track the accurate object shape evolution, while the rigid representation rather considers the global shape evolution of the object (mainly the scale and the orientation) and discards partial changes.

2.2.1 Non rigid representation

When a sufficient description of the object cannot be obtained using simple geometric elements, non rigid representation and silhouette based tracking methods are needed. An example of such requirements is shown in Figure 2.2. This modelling offers an accurate shape description based on object edges and contour. In practice this tracking scheme is used when the complete region of the object is required and offers flexibility to handle a great variety of shapes. Within this category we distinguish between two different tracking approaches, namely shape matching and contour tracking.

The shape matching approach, as any template matching technique, aims to find the image regions that show high similarity with the object model. The object silhouette is searched in the current frame based on a set of templates representing the silhouette at the previous frames. A smooth variation of the shape is assumed from a frame to the next one. The matching mechanism is mainly for-

mulated as a distance minimisation or matching score optimisation, with different metrics such as Bhattacharya distance, cross-correlation or Kullback-Leibler divergence [15].

The contour tracking approach, in contrast with the template matching, makes the contour shape evolve pixel by pixel until it fits the actual position of the object. Starting from the previous contour, the algorithms iteratively modify this contour either by expanding or shrinking some parts of it until getting the best configuration for the current frame. One major limitation is that contour tracking imperatively requires overlapping of two consecutive contours; which makes this technique unusable in case of abrupt motion. The contour evolution can be carried out using deterministic gradient descent for minimisation of a contour energy functional [121]. The tracking problem can be also considered as an estimation problem within a state space model corresponding to the shape and motion parameters of the contour [42] and solved using Kalman filters or particle filters.

2.2.2 Rigid representation

Rigid representation concerns applications where tracking accurately the object shape is not mandatory. It consists in a simpler object representation based on basic geometric forms (points, rectangles, ellipses etc.) and an easier problem statement. This representation is not restricted to objects with fixed shapes, it is also applicable to other types of object when considering only the global variation of the original shape : rotation, scale and other affine transformations. Within this category we can distinguish between two main subcategories: point representation and kernel representation.

Point tracking consists in making the correspondence between detected points of interest. In each frame the objects are represented by points provided by a detection process, and the tracker aims to match the corresponding points between successive frames to obtain the object trajectories. An example of point correspondence is shown in Figure 2.3 where the tracker computes trajectories of

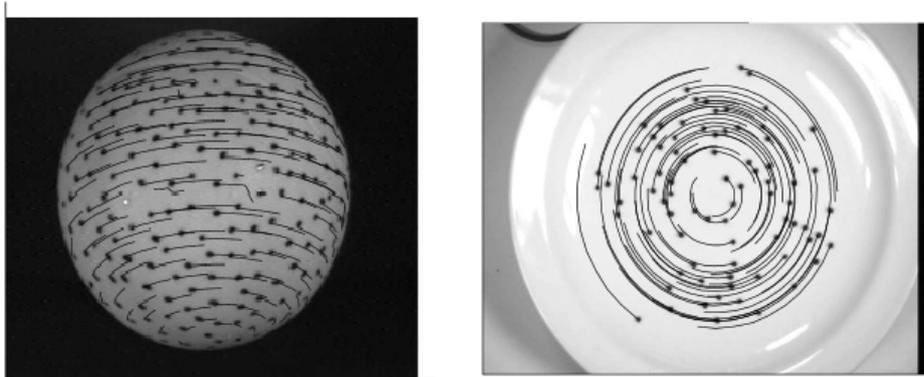


Figure 2.3: Examples of point correspondence : "Tracks obtained for a rotating ball and dish sequences." Illustration from [99].

points located on a rotating ball and a dish. This tracking scheme highly depends on the detection algorithm that returns the object location at each image frame. The correspondence problem is solved either in a deterministic way or using statistical methods.

The deterministic correspondence proceeds in two major steps: first an association cost is defined from some assumed motion constraints such as object proximity in successive frames, smooth velocity variation, group coherence when tracking groups etc. Secondly the cost is minimized to reach the optimal correspondence. This problem can be solved using several methods such as Hungarian algorithm [50], greedy searching approach [98] or using graph theoretic approach [99].

The statistical methods take into account the model uncertainties to establish the correspondence. They consider the object properties in a state space model, these properties are seen as unknown parameters to be estimated from the observation obtained from the detector. The tracker seeks then to estimate the state at each time step using the previous state and the current observation. The estimation problem can be solved using a Kalman filter, as proposed in [8] or particle filters.



Figure 2.4: Examples of kernel representations : "Patch based object representation." Illustration from [120].

Kernel tracking methods gather all cases where the object is represented with one or more kernels. These kernels can have different shapes, most of the time rectangular or elliptical, as shown in Figure 2.4. The tracking process consists in estimating the parametric motion of the kernels by maximising the appearance similarity between the object reference model and the image region at hypothesized locations. The appearance of the object can be defined using the pixels located inside the kernel, either by templates, colour histograms or other mixture models. Here also the searching mechanism can be either deterministic or statistical.

The deterministic methods search the region the most similar to the object model by a brute force mechanism in the vicinity of the previous estimated location, or by a gradient descent method such as mean-shift algorithm [16]. This algorithm iteratively increases the appearance similarity between the object model and a hypothesized object location within a predefined searching area. In the presence of occlusion, noise, similar objects,... problems of local maxima can lead to the divergence of these algorithms.

On the other hand the statistical methods model the object attributes in a state space and estimate them using the Bayesian framework. The tracking is realised, as previously, either by a Kalman filter or particle filters.

2.3 Context of the work

In this thesis, we focus on the visual tracking of objects represented by a kernel and tracking by statistical methods, more precisely by sequential Monte Carlo (SMC) methods.

The kernel representations offers the widest range of applications, as it is modular and ranges from a global modelling to a more accurate and complex object modelling: for example a human can be represented by a single rectangle or by several ellipses, one for each body part (head, torso, arms, legs etc.).

The tracking problem is formulated as a statistical inference problem and to solve this problem, we rely on sequential Monte Carlo (SMC) methods including particle filters (PFs) and sequential Markov chain Monte Carlo (MCMC) methods. These methods have gained a great interest in recent years for object tracking and have shown their capabilities to adapt to various tracking schemes. They have the advantage that they can handle complex non linear models, they offer a degree of robustness by taking into account the model uncertainties and they are easy to implement.

2.4 Chapter Conclusion

This chapter has introduced the visual tracking task and reviewed the main difficulties that still make visual tracking an open and challenging research area. A classification of the different tracking schemes has been presented. According to this classification, an object can be represented by a silhouette, a set of points or a kernel and the tracking problem can be solved by deterministic or statistical methods.

In this thesis, we have opted for a rectangular kernel based representation of the tracked object due to its wide range of applications, and for a statistical formulation of the estimation problem in the Bayesian framework, better adapted to consider the model uncertainties.

The next chapter deals with the Bayesian inference and discusses the main SMC methods, including particle filters and Markov chain Monte Carlo methods.

Chapter 3

Sequential Monte Carlo methods

This chapter focuses on Bayesian inference. First we introduce the Bayesian estimation, in particular the modelling of the estimation problem in this framework. Then we present the Monte Carlo methods, which are powerful and popular simulation based tools to solve non linear and non Gaussian problems. These methods are divided into two main families.

Particle filters (PFs) are based on importance sampling (IS). We describe the principle of IS and its adaptation to sequential problems. The weight degeneracy issue that affects sequential importance sampling is also discussed, we see how to quantify and limit this problem. Among the solutions, the focus is made on the resampling procedure and its different schemes, the choice of the importance function and the alternative PFs including the auxiliary PF and the marginal PF.

Markov chain Monte Carlo (MCMC) methods are based on the construction of a Markov chain. We present the popular Metropolis-Hastings algorithm and the adaptation of MCMC to sequential problems. Then, the choice of the proposal and the main improvements of the MCMC sampling are discussed in more detail.

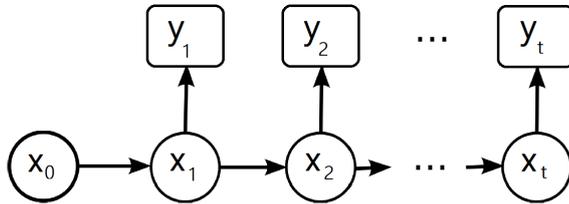


Figure 3.1: Dependence graph of a hidden Markov model.

3.1 Bayesian estimation

The Bayesian approach considers the parameters X to be estimated as random variables instead of constants and the aim is to estimate the posterior density of the parameters given the observations Y according to the Bayes rule : $p(X|Y) = \frac{p(Y|X) \cdot p(X)}{p(Y)}$. It requires the knowledge of the prior density of the parameters $p(X)$ and the likelihood $p(Y|X)$.

The Bayesian inference is suitable to address sequential applications. In this context, the problem is modelled as a Hidden Markov Model (HMM), with a discrete time t , as shown in figure 9.1. Lets assume the existence of an unobserved signal $(x_t)_{t \in \mathcal{N}}$ with $x_t \in \mathcal{R}^{n_x}$ evolving as a Markovian process described by an initial distribution $p(x_0)$ and a transition density $p(x_t|x_{t-1})$. This signal produces a sequence of observations $(y_t)_{t \in \mathcal{N}}$ with $y_t \in \mathcal{R}^{n_y}$ which are assumed to be conditionally independent given the states and to only depend on the current state x_t . This observations are described by the likelihood $p(y_t|x_t)$. The objective here is to estimate recursively the posterior distribution $p(x_{0:t}|y_{1:t})$ or the marginal distribution $p(x_t|y_{1:t})$ which is known as the filtering distribution.

In online applications such as visual tracking in the Bayesian framework, the distribution of interest is the marginal posterior $p(x_t|y_{1:t})$. Using the Bayes rule, this density is recursively expressed as follows:

$$p(x_t|y_{1:t}) \propto p(y_t|x_t) \cdot p(x_t|y_{1:t-1}) \tag{3.1}$$

with :

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1}) \cdot p(x_{t-1}|y_{1:t-1}) dx_{t-1}$$

The prior density $p(x_t|x_{t-1})$ represents the dynamic evolution of the state x_t given the previous state x_{t-1} , and the observation likelihood $p(x_t|y_{1:t})$ measures the matching accuracy of the observation y_t given the state x_t .

Finally the hidden state is estimated from the posterior distribution by minimising the *mean square error* (MMSE estimator), as in equation 3.2, or by maximising the *a posteriori* probability (MAP estimator) as in equation 3.3.

$$\hat{x}_t^{MMSE} = E_{x|y}[x_t] = \int x_t \cdot p(x_t|y_{1:t}) \cdot dx_t \quad (3.2)$$

$$\hat{x}_t^{MAP} = \arg \max_x p(x_t|y_{1:t}) \quad (3.3)$$

When the HMM is linear and Gaussian, the filtering problem can be solved analytically using exact inference. In this case broadly known techniques have shown their ability to deal with this problem such as Kalman filters. However, these simple models correspond to limited scenarios and do not fit most real life applications. In these cases the problem cannot be solved analytically, it can only be handled by using approximation techniques. To cope with non linear and non Gaussian problems, we use the sequential Monte Carlo (SMC) methods, which are powerful simulation based methods that enable to approximate the posterior distribution.

The principle of SMC methods is to represent the target distribution by a set of random samples with associated weights and to compute the estimates from these samples and weights. They can be divided into two main families : the particle filters (PFs) and the sequential MCMC methods.

3.2 Particle Filtering

In the Bayesian filtering domain, among the SMC methods, particle filtering (PF) is one of the most popular methods. Particle filtering aims to construct a discrete

3. Sequential Monte Carlo methods

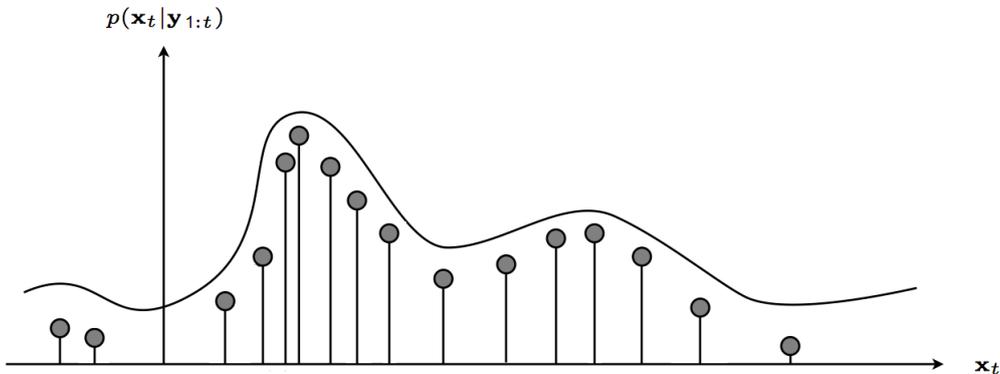


Figure 3.2: Particular approximation of a density $p(x_t|y_{1:t})$

approximation of the posterior density from a set of N_p samples $\{x_{0:t}^{(i)}\}_{i=1}^{N_p}$, called particles, with associated probabilities, called weights, $\{\pi_t^{(i)}\}_{i=1}^{N_p}$.

The posterior distribution is approximated as:

$$\hat{p}(x_{0:t}|y_{1:t}) = \sum_{i=1}^{N_p} \pi_t^{(i)} \cdot \delta(x_{0:t} - x_{0:t}^{(i)})$$

Figure 3.2 illustrates the particle approximation of a density.

If we are only interested in the filtering distribution $p(x_t|y_{1:t})$, we marginalise the samples over the previous states $x_{0:t-1}$.

In the case we have the capability to sample N_p independent and identically distributed (i.i.d.) samples from the posterior $p(x_{0:t}|y_{1:t})$, we notice that the particle related probabilities are $\{\pi_t^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$. Therefore the posterior is straightforwardly approximated as:

$$\hat{p}(x_{0:t}|y_{1:t}) = \frac{1}{N_p} \cdot \sum_{i=1}^{N_p} \delta(x_{0:t} - x_{0:t}^{(i)})$$

This approximation converges to the target density when the number of particles tends to infinity.

3. Sequential Monte Carlo methods

Unfortunately the posterior is generally not available to draw samples. Therefore alternative sampling schemes are needed to generate the samples and calculate the associated weights. One of these techniques is importance sampling.

3.2.1 Importance sampling

Importance sampling is a general Monte Carlo approach that handles cases where sampling directly from the posterior distribution is impossible. An importance function $q(x_{0:t}|y_{1:t})$ is used to easily sample and generate N_p i.i.d samples $\{x_{0:t}^{(i)}\}_{i=1}^{N_p}$. Then the associated weights are obtained according to the importance sampling principle:

$$w(x_{0:t}^{(i)}) = \frac{p(x_{0:t}^{(i)}|y_{1:t})}{q(x_{0:t}^{(i)}|y_{1:t})}$$

Finally, we get the following discrete approximation of the posterior:

$$\hat{p}(x_{0:t}|y_{1:t}) = \sum_{i=1}^{N_p} w_t^{(i)} \cdot \delta(x_{0:t} - x_{0:t}^{(i)})$$

where $w_t^{(i)}$ are the normalised importance weights:

$$w_t^{(i)} = \frac{w(x_{0:t}^{(i)})}{\sum_{i=1}^{N_p} w(x_{0:t}^{(i)})}$$

Importance sampling aims to estimate the whole sequence (or trajectory) of the state $x_{0:t}$ and the importance weights which represent the adequateness between the whole proposed trajectory and the sequence of observations. In this form, importance sampling is not adequate for recursive estimation: at each iteration, when new observations are available, the whole trajectory $x_{0:t}$ needs to be drawn again and the weights of the whole trajectory need to be recomputed. This limitation has been overcome by the development of the sequential importance sampling algorithm.

3.2.2 Sequential importance sampling (SIS)

SIS is a modified importance sampling method that enables to sample from $q(x_{0:t}|y_{1:t})$ without modifying the previously simulated trajectories (i.e. the past samples $x_{0:t-1}$). For that purpose, we consider an importance function which can be written recursively as:

$$q(x_{0:t}|y_{1:t}) = q(x_{0:t-1}|y_{1:t-1}) \cdot q(x_t|x_{0:t-1}, y_{1:t})$$

Each particle $x_{0:t}^{(i)}$ is obtained by adding a new state $x_t^{(i)}$ sampled from $q(x_t|x_{0:t-1}^{(i)}, y_{1:t})$. Then we get a recursive form of the associated importance weights:

$$w_t^{(i)} \propto w_{t-1}^{(i)} \cdot \frac{p(y_t|x_t^{(i)}) \cdot p(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t})} \quad (3.4)$$

The weights are normalised to get $\sum_{i=1}^{N_p} w_t^{(i)} = 1$.

Finally the approximation of the posterior is:

$$\hat{p}(x_{0:t}|y_{1:t}) = \sum_{i=1}^{N_p} w_t^{(i)} \cdot \delta(x_{0:t} - x_{0:t}^{(i)}).$$

The filtering density is obtained by marginalising the posterior approximation:

$$\hat{p}(x_t|y_{1:t}) = \int \hat{p}(x_{0:t}|y_{1:t}) \cdot dx_{0:t-1} = \sum_{i=1}^{N_p} w_t^{(i)} \cdot \delta(x_t - x_t^{(i)}) \quad (3.5)$$

The MMSE estimate of the hidden state x_t is:

$$\hat{x}_t = \sum_{i=1}^{N_p} x_t^{(i)} \cdot w_t^{(i)}$$

Finally this algorithm operates in two steps: a sampling step where the particles are propagated from the importance function, and a weighting step where the adequateness of these particles with the observation is evaluated. The algorithm is synthesized in Table 3.1.

<p>Initialisation (t=0) sample $\{x_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$, $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Sequential processing (t>0) for $i = 1 : N_p$ do sample $x_t^{(i)} \sim q(x_t x_{0:t-1}^{(i)}, y_{1:t})$ set $x_{0:t}^{(i)} = (x_{0:t-1}^{(i)}, x_t^{(i)})$ evaluate $w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{p(y_t x_t^{(i)}) \cdot p(x_t^{(i)} x_{t-1}^{(i)})}{q(x_t^{(i)} x_{0:t-1}^{(i)}, y_{1:t})}$ end for normalise the importance weights</p>

Table 3.1: SIS particle filter algorithm.

3.2.3 The weight degeneracy problem

The problem of weight degeneracy is one of the major limitations of the importance sampling method. In practice the variance of the importance weights increases over time. After a few iterations of the SIS algorithm, only a few particles have significant weights, all the other particles have weights close to zero and their contribution to the final estimate is almost zero [10]. This leads to a poor estimate of the filtering density.

In order to quantify this weight degeneracy problem, several testing criteria have been proposed. Authors in [48] have proposed the coefficient of variation (CV) of the normalised weights defined by :

$$CV = \sqrt{\frac{1}{N_p} \cdot \sum_{i=1}^{N_p} (N_p w^{(i)} - 1)^2} \quad (3.6)$$

This indicator shows how uneven the normalised importance weights are. The most used indicator, proposed by [60], is the effective sample size (ESS) that explicitly indicates the number of effective particles, i.e. the number of particles

3. Sequential Monte Carlo methods

that contribute significantly to the estimate. It is defined by:

$$ESS = \frac{1}{\sum_{i=1}^{N_p} (w^{(i)})^2} \quad (3.7)$$

A smaller value of the ESS means a larger variance of the weights and more weight degeneracy.

To reduce the weight degeneracy, several methods have been proposed. The concept of resampling has been first introduced by Gordon et al.(1993) and several works, since then, have been devoted to propose effective resampling schemes. The main resampling schemes are discussed in the next section.

Another crucial element to limit the degeneracy of the algorithm, is the importance function itself. The choice of the importance function is of high interest as it conditions the relevance of the proposed samples, and has a strong impact on the quality of the Monte Carlo estimation. Indeed if the importance function is well chosen, the particles are propagated in the regions of interest of the state space, the associated importance weights have similar high values and the variance remains low, which avoids weight degeneracy.

Moreover, alternative PFs have been developed to reduce the weight degeneracy. Among them, the auxiliary PF performs resampling by considering the future observation and the marginal PF aims to sample directly from the filtering density. Both algorithms are introduced later.

3.2.4 Resampling

The main idea behind resampling is to generate a new set of particles from the current population by replacement to reduce the weight variance of the set. This is realised by duplicating particles having high weight values to replace the particles with low weights. The resampling process is mainly based on the importance weights and tends to reduce their discrepancy.

To keep a correct number of effective particles and maintain the stability of the Monte Carlo estimate for long time running applications, resampling must be done several times. In some cases, such as in the bootstrap filter, resampling is realised at each time step, but it is not always a suitable choice. Indeed resam-

3. Sequential Monte Carlo methods

pling has two major drawbacks, first it adds a computational cost and reduces the algorithmic performance of the filter. Secondly, resampling introduces additional variance in the Monte Carlo approximation [10]. Duplicating the highly weighted particles leads to a loss of particle diversity which reduces the accuracy of the Monte Carlo approximation. This problem is known as sample impoverishment. This is why it is preferable to perform resampling only when a significant weight degeneracy is observed. For that purpose a resampling routine is triggered whenever the ESS or any other indicator falls below a threshold.

To perform resampling, several methods have been developed. In this work we focus on the most widely used schemes, which respect two constraints: keep the number of particles constant and make the importance weights equal after resampling. They duplicate the particles a number of times in proportion to their importance weights. A more detailed analysis and performance study on resampling algorithms is given in [38] and [21].

Multinomial resampling

It is the simplest and most intuitive scheme of resampling [21]. It consists in duplicating the particles $x^{(I)}$ where I is a set of indexes obtained by sampling from the multinomial distribution $\text{Mult}(N_p; w^{(1)}, \dots, w^{(N_p)})$. This method is simple to implement and still sufficient for resampling purpose. However its major drawback is the computational cost since we have to sort the importance weights and compare them with the uniformly drawn variable to get the appropriate index. This process may need at best $N_p \log(N_p)$ operations [10].

Residual resampling

This resampling scheme is called residual because it resamples only a residual part of the particle population. This resampling scheme proposed by [61] operates in two steps. First it allocates $N' = \sum_{i=1}^{N_p} n'_i$ particles, by duplicating each particle $x^{(i)}$ n'_i times with $n'_i = \lfloor N_p \cdot w^{(i)} \rfloor$. The remaining "non allocated" part of the particle set $N'' = N_p - N'$ is obtained then by performing multinomial resampling (or any other scheme) on the whole particle set using the new weights $\bar{w}^{(i)} = N_p \cdot w^{(i)} - n'_i$. This method permits to significantly reduce the

3. Sequential Monte Carlo methods

computational cost of the resampling process, since a significant part is done by allocation and only the reminding part is realised by sorting and sampling from a distribution. Therefore instead of $N_p \log(N_p)$ operations, this amount is reduced to $N \log(N)$ operations.

Stratified resampling

Stratification is a method originated from survey sampling and consists in subdividing the $[0, 1]$ interval into N_p multiple disjoint sub-intervals. Next N_p ordered independent variables $u^i \sim \mathcal{U}[(i-1)/N_p, i/N_p]$ are drawn and used to select the particles using the inversion method, as in multinomial sampling. Stratified resampling is quite similar to multinomial resampling, as it only introduces a different way of generating the sequence of indexes. However, this resampling scheme is shown to enhance the resampling quality, due to the partitioning which clearly reduces the discrepancy of the set of samples u^i from the uniform distribution [10]. The stratified resampling is also shown to reduce the computational cost compared to multinomial resampling [38].

Systematic resampling

First introduced by Carpenter et al. 1999 [12], the systematic resampling, or universal sampling, is an extension of the stratified resampling. The idea is to pursue the discrepancy reduction one step further, by making the variables u^i dependent and equidistant from each other. Unlike the stratified resampling, the systematic resampling draws only one variable $u^1 \sim \mathcal{U}[0, 1/n]$ and sets the other variables $u^i = u^1 + (i-1)/N_p$. As for the stratified resampling the inversion method is still used here to obtain our particle set. As it reduces even more the discrepancy of the set of uniformly distributed variables u^i and needs only a single draw, the systematic resampling performs as good as the stratified resampling with a lower computational cost [38]. However, no theoretical proof can be presented to support the variance minimisation due to the fact that the variables u^i are dependent.

These are the main resampling schemes that are widely used and embedded in particle filtering algorithms to counter the weight degeneracy issues. In this thesis

3. Sequential Monte Carlo methods

we use the residual resampling scheme, as it is simple to implement and offers satisfactory results. Also in our implementations we have fixed the threshold to launch the resampling procedure at $N_p/3$.

The resulting algorithm, known as sequential importance resampling (SIR) particle filter, is shown in Table 3.2.

<p>Initialisation (t=0) sample $\{x_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$, $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Sequential processing (t>0) for $i = 1 : N_p$ do sample $x_t^{(i)} \sim q(x_t x_{0:t-1}^{(i)}, y_{1:t})$ set $x_{0:t}^{(i)} = (x_{0:t-1}^{(i)}, x_t^{(i)})$ evaluate $w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{p(y_t x_t^{(i)}) \cdot p(x_t^{(i)} x_{t-1}^{(i)})}{q(x_t^{(i)} x_{0:t-1}^{(i)}, y_{1:t})}$ end for normalise the importance weights if ($ESS < \frac{N_p}{3}$) resample end if</p>

Table 3.2: SIR particle filter algorithm.

3.2.5 Choice of the importance function

The choice of the importance function is of crucial importance for the success of the particle filtering process. A well chosen importance function ensures drawing particle sets with low weight variance. It also allows to perform a satisfactory exploration of the state space with a reasonable number of particles.

In most cases, the transition density, also known as the prior density and related to the *a priori* evolution model, is used as the importance function. This choice is made in a great number of particle filters, in particular in the Bootstrap filter, leading to a simple weight update $w_t^{(i)} \propto w_{t-1}^{(i)} \cdot p(y_t | x_t^{(i)})$. The Bootstrap filter systematically uses a resampling routine to avoid weight degeneracy. The Bootstrap filter algorithm is synthesized in Table 3.3.

3. Sequential Monte Carlo methods

<p>Initialisation (t=0) sample $\{x_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$, $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Sequential processing (t>0) for $i = 1 : N_p$ do sample $x_t^{(i)} \sim p(x_t x_{t-1}^{(i)})$ set $x_{0:t}^{(i)} = (x_{0:t-1}^{(i)}, x_t^{(i)})$ evaluate $w_t^{(i)} = p(y_t x_t^{(i)})$ end for normalise the importance weights resample</p>

Table 3.3: Bootstrap filter algorithm.

However this choice can lead to bad estimation results, especially when the evolution model is imprecise or unpredictable. As the prior density is not very informative and does not take into account the knowledge about the observations, sampling using the prior corresponds to a blind exploration of the state space, and can generate particles in inappropriate regions.

The best choice in the sense of weight variance minimisation is the optimal importance function [22] which takes into account the latest observation: $p(x_t|x_{t-1}, y_t) = \frac{p(y_t|x_t) \cdot p(x_t|x_{t-1})}{p(y_t|x_{t-1})}$. This function minimizes the conditional variance of the importance weights given the simulated trajectory $x_{0:t-1}^{(i)}$ and the observations $y_{1:t}^{(i)}$. As a matter of fact, the importance weights are then written as $w_t^{(i)} \propto w_{t-1}^{(i)} \cdot p(y_t|x_{t-1}^{(i)})$. They no longer depend on the current particles $x_t^{(i)}$, so $\text{var}(w_t^{(i)}|x_{0:t-1}^{(i)}, y_{1:t}^{(i)}) = 0$. Unfortunately in most applications, computing and sampling from the optimal proposal is impossible because the analytic form is unavailable or calculations are prohibitive.

Several suboptimal methods, aiming to offer a proposal close to the optimal importance function, have been proposed. These suboptimal strategies can be divided into two main categories. The explicit methods, also known as data driven methods, focus on the design of effective importance functions from the available information on the observations. The implicit methods use the prior density as the proposal and develop an additional optimisation step to move the

particles toward the most likely areas of the state space.

These methods are reviewed in Chapter 5 dedicated to the optimal proposal and its approximation.

In the next section we discuss the main alternative particle filtering algorithms that try to reduce the weight degeneracy.

3.2.6 Auxiliary particle filter (APF)

In the basic PF, resampling is performed at the end of an iteration (or time step) just before taking into account the next observation. Intuitively, it seems more efficient to resample at the beginning of an iteration.

The main idea behind the auxiliary PF is to perform resampling in the light of the next observation, that is to exploit the knowledge about the next observation to determine which particles should survive resampling and should be propagated in the state space. The auxiliary particle filter proposed by Pitt and Shephard (1999)[81] is an extension of the particle filtering algorithm, based on the addition of an auxiliary variable to help the simulation task.

The APF is also based on the importance sampling principle but the target distribution is no longer the filtering density (9.1), but an approximation derived from the particle approximation (3.5) of the previous filtering density $p(x_{t-1}|y_{1:t-1})$:

$$\hat{p}(x_t|y_{1:t}) \propto p(y_t|x_t) \cdot \sum_{i=1}^{N_p} w_{t-1}^{(i)} \cdot p(x_t|x_{t-1}^{(i)}) \quad (3.8)$$

As $p(y_t|x_t) \cdot p(x_t|x_{t-1}) = p(y_t|x_{t-1}) \cdot p(x_t|x_{t-1}, y_t)$, this approximation can be rewritten as:

$$\hat{p}(x_t|y_{1:t}) \propto \sum_{i=1}^{N_p} w_{t-1}^{(i)} \cdot p(y_t|x_{t-1}^{(i)}) \cdot p(x_t|x_{t-1}^{(i)}, y_t) \quad (3.9)$$

This mixture can be seen as the marginal distribution of a joint distribution:

$$p(x_t, k|y_{1:t}) \propto w_{t-1}^{(k)} \cdot p(y_t|x_{t-1}^{(k)}) \cdot p(x_t|x_{t-1}^{(k)}, y_t) \quad (3.10)$$

3. Sequential Monte Carlo methods

with $p(k|y_{1:t}) \propto w_{t-1}^{(k)} \cdot p(y_t|x_{t-1}^{(k)})$ and $p(x_t, k|y_{1:t}) = p(x_t|x_{t-1}^{(k)}, y_t)$.

The dimension of the problem is extended by adding the auxiliary variable k which represents an index on the mixture defined from the previous particles and allows to select the most relevant particles according to y_t .

To sample from this distribution, the APF uses a proposal of the same form :

$$q(x_t, k|y_{1:t}) = q(k|y_{1:t}) \cdot q(x_t|y_{1:t}, k)$$

where

$$q(k|y_{1:t}) = \lambda_{t-1}^{(k)} = \frac{w_{t-1}^{(k)} \cdot p(y_t|x_{t-1}^{(k)})}{\sum_{i=1}^{N_p} w_{t-1}^{(i)} \cdot p(y_t|x_{t-1}^{(i)})} \quad (3.11)$$

is called the first stage weight, and used to draw an index k and $q(x_t|y_{1:t}, k) = q(x_t|x_{t-1}^{(k)}, y_{1:t})$ is a generic importance function used to propagate a new particle from the previous particle of index k .

The sampling process is performed in two stages, first the indexes are obtained according to the first stage weights and then, the candidates for time t are sampled by propagating the preselected particles. This process operates a resampling implicitly: indeed, this two step sampling is equivalent to resample $\{x_{t-1}^{(k)}\}_{k=1}^{N_p}$ according to $\lambda_{t-1}^{(k)}$ and propose candidates according to the new set of particles.

The importance weights are obtained according to the importance sampling principle:

$$w_t^{(k)} \propto \frac{p(y_t|x_t^{(k)}) \cdot p(x_t^{(k)}|x_{t-1}^{(k)})}{\hat{p}(y_t|x_{t-1}^{(k)}) \cdot q(x_t^{(k)}|x_{t-1}^{(k)}, y_{1:t})} \quad (3.12)$$

The resulting auxiliary particle filter algorithm is synthesised in Table 3.4.

<p>Initialisation (t=0) sample $\{x_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$, $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Filtering at time t for $i = 1 : N_p$ do evaluate the first stage weights $\lambda_{t-1}^{(i)} = w_{t-1}^{(i)} \cdot \hat{p}(y_t x_{t-1}^{(i)})$ end for normalise the first stage weights $\lambda_{t-1}^{(i)}$ resample $\{x_{t-1}^{(i)}, \lambda_{t-1}^{(i)}\}_{i=1}^{N_p}$ to obtain $\{x_{t-1}^{(i)}, \frac{1}{N_p}\}_{i=1}^{N_p}$ for $i = 1 : N_p$ do sample $x_t^{(i)} \sim q(x_t x_{t-1}^{(i)}, y_{1:t})$ evaluate $w_t^{(i)} = \frac{p(y_t x_t^{(i)}) \cdot p(x_t^{(i)} x_{t-1}^{(i)})}{\hat{p}(y_t x_{t-1}^{(i)}) \cdot q(x_t x_{t-1}^{(i)}, y_{1:t})}$ end for normalise the importance weights</p>

Table 3.4: Auxiliary particle filter algorithm.

The performance of the APF depends on the choice of the first stage weights, more precisely of the approximation of the predictive likelihood $\hat{p}(y_t|x_{t-1}^{(k)})$, and of the importance function $q(x_t|x_{t-1}^{(k)}, y_{1:t})$.

The most popular choice is to set $\hat{p}(y_t|x_{t-1}^{(k)}) = p(y_t|\mu_t^{(k)})$ where $\mu_t^{(k)}$ is the mode, mean or median of the prior distribution $p(x_t|x_{t-1}^{(k)})$ and to set $q(x_t|x_{t-1}^{(k)}, y_{1:t}) = p(x_t|x_{t-1}^{(k)})$. But this simple approximation is not optimal and can lead to estimates with large variance.

In the terminology of Pitt and Shephard, an APF which employs the exact predictive likelihood $p(y_t|x_{t-1}^{(k)})$ and the optimal importance function $p(x_t|x_{t-1}^{(k)}, y_{1:t})$ is called fully adapted (FA). In this case, the weights $w_t^{(k)}$ are all equal.

3.2.7 Marginal particle filter (MPF)

The basic PF approximates the joint posterior $p(x_{0:t}|y_{1:t})$ from the approximation of $p(x_{0:t-1}|y_{1:t-1})$ by adding a new sample x_t to each particle $x_{0:t-1}$. So the dimension of the state space grows at each iteration, which results in a larger variance of the importance weights.

3. Sequential Monte Carlo methods

In many applications using sequential Monte Carlo methods, observations are collected on the fly and we are only interested in estimating the marginalised filtering distribution $p(x_t|y_{1:t})$. It seems not efficient to marginalise the joint posterior to obtain this distribution. The marginal PF performs filtering via simulation directly from the marginal distribution, instead of the joint posterior.

More precisely, the target distribution is the approximation of $p(x_t|y_{1:t})$ obtained from the particle estimation $\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_p}$ of the previous marginal $p(x_{t-1}|y_{1:t-1})$:

$$\hat{p}(x_t|y_{1:t}) \propto p(y_t|x_t) \cdot \sum_{i=1}^{N_p} w_{t-1}^{(i)} \cdot p(x_t|x_{t-1}^{(i)}) \quad (3.13)$$

To sample from this distribution with the MPF, the choice of the proposal is free for any density that is suitable for the application. But it's convenient to choose a proposal with a similar form [47] :

$$q(x_t|y_{1:t}) = \sum_{j=1}^{N_p} w_{t-1}^{(j)} \cdot q(x_t|x_{t-1}^{(j)}, y_t)$$

where $q(x_t|x_{t-1}^{(j)}, y_t)$ is a generic importance function.

The marginal importance weights are expressed as:

$$w_t = \frac{\hat{p}(x_t|y_{1:t})}{q(x_t|y_{1:t})}$$

For a particle $x_t^{(i)}$, they are evaluated as:

$$w_t^{(i)} = \frac{p(y_t|x_t^{(i)}) \cdot \sum_{j=1}^{N_p} w_{t-1}^{(j)} \cdot p(x_t^{(i)}|x_{t-1}^{(j)})}{q(x_t^{(i)}|y_{1:t})} \quad (3.14)$$

This expression of the marginal weights highlights the interest of using the MPF for filtering. Indeed, unlike the SIS weight expressed in (3.4) and obtained by updating $w_{t-1}^{(i)}$, which results in a weight variance increasing from a time to another, in the MPF the weight at time t is computed from scratch, directly from the ratio between the target and the proposal. For filtering applications, the MPF is known to reduce the importance weight variance over the standard SIS [47], but

it has a higher computational cost.

The resulting MPF is synthesized in Table 3.5.

<p>Initialisation (t=0) sample $\{x_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$, $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Filtering at time t for $i = 1 : N_p$ do sample $x_t^{(i)} \sim \sum_{j=1}^{N_p} w_{t-1}^{(j)} \cdot q(x_t x_{t-1}^{(j)}, y_t)$ evaluate $w_t^{(i)} = \frac{p(y_t x_t^{(i)}) \cdot \sum_{j=1}^{N_p} w_{t-1}^{(j)} \cdot p(x_t^{(i)} x_{t-1}^{(j)})}{\sum_{j=1}^{N_p} w_{t-1}^{(j)} \cdot q(x_t^{(i)} x_{t-1}^{(j)}, y_t)}$ end for normalise importance weights $w_t^{(i)}$</p>

Table 3.5: Marginal particle filter algorithm.

The performance of the MPF depends on the choice of the proposal and more precisely of $q(x_t|x_{t-1}^{(j)}, y_t)$. We can notice that if $q(x_t|x_{t-1}^{(j)}, y_t) = p(x_t|x_{t-1}^{(j)})$, then $w_t^{(i)} = p(y_t|x_t^{(i)})$ and the MPF is equivalent to the Bootstrap PF.

The best choice is to set $q(x_t|x_{t-1}^{(j)}, y_t) = p(x_t|x_{t-1}^{(j)}, y_t)$ but the optimal importance function is generally unavailable.

3.3 Sequential Markov chain Monte Carlo Methods

Markov chain Monte Carlo (MCMC) methods are an alternative to the importance sampling methods to perform Bayesian estimation via simulation. They are known to be more efficient in high dimensional state spaces. The objective of these methods is to construct an ergodic Markov chain that converges to the target distribution, which is the posterior distribution in the Bayesian framework. The construction of the Markov chain is done iteratively by producing each new sample from the previous one, using a Markov kernel with an invariant distribution equal to the target density.

3.3.1 MCMC methods

The two popular algorithms to generate Markov chains are the Metropolis-Hastings algorithm [67, 37] and the Gibbs sampler [30]. In this work we focus only on the Metropolis-Hastings (MH) algorithm which is more generic since it imposes minimal requirements on the target density. The Gibbs sampler requires to derive some conditional densities from the target density.

Let π be the target distribution and $x^{(i-1)}$ the sample of the Markov chain at iteration $i - 1$. The MH algorithm consists of two steps. First a proposal density $q(x|x^{(i-1)})$ is defined and used to easily generate a new sample x^* . Then this sample is accepted or rejected according to the accept-reject principle, which corrects the fact that the proposal density is not the target distribution. The acceptance probability or ratio is defined by:

$$\alpha = \min \left(1, \frac{\pi(x^*) \cdot q(x^{(i-1)}|x^*)}{\pi(x^{(i-1)}) \cdot q(x^*|x^{(i-1)})} \right) \quad (3.15)$$

To sum up, to obtain a Markov chain $\{x^{(i)}\}_{i \geq 0}$ that converges to an invariant distribution π , the generic MCMC algorithm developed by Metropolis et al. is given in Table 3.6.

<p>Initialisation (i=0) Initialise the Markov chain with $x^{(0)} \sim q(x)$</p>
<p>Iteration i sample $x^* \sim q(x x^{(i-1)})$ compute the acceptance ratio $\alpha = \min \left(1, \frac{\pi(x^*) \cdot q(x^{(i-1)} x^*)}{\pi(x^{(i-1)}) \cdot q(x^* x^{(i-1)})} \right)$ $x^{(i)} = \begin{cases} x^* & \text{with probability } \alpha \\ x^{(i-1)} & \text{with probability } (1 - \alpha) \end{cases}$</p>

Table 3.6: Metropolis-Hastings algorithm.

The process is similar to a stochastic optimization algorithm. The algorithm always accepts values x^* such that the ratio $\frac{\pi(x^*)}{q(x^*|x^{(i-1)})}$ is increased compared to its previous value, else it accepts x^* such that the ratio is decreased with the

3. Sequential Monte Carlo methods

acceptance probability α .

In the special case of a symmetric proposal density $q(x^*|x^{(i-1)}) = q(x^{(i-1)}|x^*)$, the acceptance rate reduces to:

$$\alpha = \min \left(1, \frac{\pi(x^*)}{\pi(x^{(i-1)})} \right) \quad (3.16)$$

This case is common since the most popular proposal is the Gaussian random walk proposal, that is a normal density centred at the current state of the Markov chain $x^{(i-1)}$.

Regardless the initial value $x^{(0)}$ and the proposal, the Markov chain generated according to the MH algorithm converges to the desired target density π . The first samples of the chain are discarded as a burn-in period is necessary for the Markov chain to achieve its steady state.

Finally the N_p last samples or particles of the Markov chain are extracted to get the approximation of the target distribution :

$$\hat{\pi}(x) = \frac{1}{N_p} \sum_{i=N_b+1}^{N_b+N_p} \delta(x - x^{(i)}) \quad (3.17)$$

The choice of the burn-in period N_b and of the chain length $N_b + N_p$ is empirically determined. We can notice that the Markov chain may remain in the same state, so the samples are not independent and may be highly correlated.

3.3.2 Sequential MCMC

The MCMC sampling scheme, as stated previously, iteratively constructs a Markov chain that converges to a time invariant target density. To address time evolving applications, we can directly run several MCMC routines, one for each time step. However in Bayesian filtering problems, the target density is the marginal posterior distribution $p(x_t|y_{1:t})$, which is analytically intractable. So it is impossible

3. Sequential Monte Carlo methods

to compute the acceptance ratio which is written as:

$$\alpha = \min \left(1, \frac{p(x_t^*|y_{1:t}) \cdot q(x_t^{(i-1)}|x_t^*)}{p(x_t^{(i-1)}|y_{1:t}) \cdot q(x_t^*|x_t^{(i-1)})} \right) \quad (3.18)$$

The solution consists in using the recursive expression (9.1) of $p(x_t|y_{1:t})$ and the MCMC sampling scheme in a sequential setting. Several sequential variants of MCMC have been proposed in the literature to solve online filtering problems.

The sequential MCMC method proposed by Khan et al. [46] is the most commonly used method. The key idea is to replace the filtering density by its approximation derived from the unweighted particle approximation of the previous density $p(x_{t-1}|y_{1:t-1})$. This approximation is based on the N_p last samples of the Markov chain generated at the previous time $\{x_{t-1}^{(i)}\}_{i=1}^{N_p}$.

By replacing in (9.1), we get:

$$\hat{p}(x_t|y_{1:t}) \propto \frac{1}{N_p} \cdot p(y_t|x_t) \cdot \sum_{i=1}^{N_p} p(x_t|x_{t-1}^{(i)})$$

The sequential MCMC algorithm of Khan et al. [46] is described in Table 3.7. It corresponds to the Metropolis-Hastings algorithm with a target distribution equal to $\hat{p}(x_t|y_{1:t})$.

<p>At time t</p> <p>Initialisation (i=1)</p> <p>randomly select a sample $x_{t-1}^{(r)}$ and initialise the Markov chain</p> <p>$x_t^{(1)} \sim p(x_t x_{t-1}^{(r)})$</p> <hr/> <p>for $i = 2 : N_b + N_p$ do</p> <div style="margin-left: 20px;"> <p>propose a move $x_t^* \sim q(x_t x_t^{(i-1)})$</p> <p>compute the acceptance ratio $\alpha = \min \left(1, \frac{\hat{p}(x_t^* y_{1:t}) \cdot q(x_t^{(i-1)} x_t^*)}{\hat{p}(x_t^{(i-1)} y_{1:t}) \cdot q(x_t^* x_t^{(i-1)})} \right)$</p> <p>accept $x_t^{(i)} = x_t^*$ with probability α, else set $x_t^{(i)} = x_t^{(i-1)}$</p> </div> <p>end for</p> <p>keep the N_p last samples after a burn-in period $\{x_t^{(i)}\}_{i=1}^{N_p} = \{x_t^{(i)}\}_{i=N_b+1}^{N_b+N_p}$</p>

Table 3.7: Sequential MCMC algorithm.

3. Sequential Monte Carlo methods

The major drawback of this method is the computational cost of the predictive density. This cost increases linearly with the number of particles used for the Monte Carlo approximation.

Once the Markov chain is generated at time t , the N_p last samples are extracted and re-indexed from 1 to N_p to obtain the approximation of the filtering density:

$$\hat{p}(x_t|y_{1:t}) \propto \frac{1}{N_p} \cdot \sum_{i=1}^{N_p} \delta(x_t - x_t^{(i)})$$

To take into account the time evolution of the state between two times in the sampling step, each Markov chain is initialised according to the dynamic model. A sample is randomly selected in the previous particle set $\{x_{t-1}^{(i)}\}_{i=1}^{N_p}$ and the first sample of the chain at time t is generated from the transition or prior density $p(x_t|x_{t-1})$. Then to propose the other candidates, we can choose the classical symmetric proposal, commonly used in MCMC algorithms, which performs a local move around the previous sample of the Markov chain $x_t^{(i-1)}$ or an independent proposal which is the prior density $q(x_t|x_t^{(i)}) = p(x_t|x_{t-1})$. Then the proposed sample is independent of the previous sample of the chain.

An alternative algorithm is the joint sequential MCMC algorithm proposed by Septier et al. [95, 75]. To avoid the numerical integration of the predictive density, this MCMC algorithm targets the joint posterior distribution $p(x_t, x_{t-1}|y_{1:t}) \propto p(y_t|x_t) \cdot p(x_t|x_{t-1}) \cdot p(x_{t-1}|y_{1:t-1})$. By using the particle approximation of the previous filtering density $p(x_{t-1}|y_{1:t-1})$, this joint density can be approached by:

$$\hat{p}(x_t, x_{t-1}|y_{1:t}) \propto p(y_t|x_t) \cdot p(x_t|x_{t-1}) \cdot \sum_{i=1}^{N_p} \delta(x_{t-1} - x_{t-1}^{(i)})$$

The algorithm is described in Table 3.8. It corresponds to the Metropolis-Hastings algorithm with a target distribution equal to $\hat{p}(x_t, x_{t-1}|y_{1:t})$.

In the sampling step, the joint sequential MCMC algorithm uses a joint proposal with the following form:

3. Sequential Monte Carlo methods

$$q(x_t, x_{t-1} | x_t^{(i-1)}, x_{t-1}^{(i-1)}) \propto q(x_t | x_{t-1}, x_t^{(i-1)}) \cdot \sum_{j=1}^{N_p} \delta(x_{t-1} - x_{t-1}^{(j)})$$

This form leads to a simplification in the acceptance ratio α .

First a sample x_{t-1}^* is selected from a discrete uniform distribution on the set of particles $\{x_{t-1}^{(i)}\}_{i=1}^{N_p}$ obtained at the previous time $t - 1$. Then a sample x_t^* is generated from x_{t-1}^* using a proposal $q(x_t | x_{t-1}, x_t^{(i-1)})$. If the prior density $p(x_t | x_{t-1})$ is used, the acceptance ratio simply reduces to:

$$\alpha = \min \left(1, \frac{p(y_t | x_t^*)}{p(y_t | x_t^{(i-1)})} \right) \quad (3.19)$$

As we are only interested in estimating the marginal posterior, the joint posterior is marginalised upon x_{t-1} and the particles $\{x_{t-1}^{(i)}\}_{i=1}^{N_p}$ are discarded.

<p>At time t</p> <p>Initialisation</p> <p>randomly select a sample $x_{t-1}^{(1)}$ and initialise the Markov chain</p> $x_t^{(1)} \sim p(x_t x_{t-1}^{(1)})$
<p>for $i = 2 : N_b + N_p$ do</p> <div style="margin-left: 20px;"> <p>propose a move $(x_t^*, x_{t-1}^*) \sim q(x_t, x_{t-1} x_t^{(i-1)}, x_{t-1}^{(i-1)})$</p> <p>compute the acceptance ratio</p> $\alpha = \min \left(1, \frac{\hat{p}(x_t^*, x_{t-1}^* y_{1:t}) \cdot q(x_t^{(i-1)}, x_{t-1}^{(i-1)} x_t^*, x_{t-1}^*)}{\hat{p}(x_t^{(i-1)}, x_{t-1}^{(i-1)} y_{1:t}) \cdot q(x_t^*, x_{t-1}^* x_t^{(i-1)}, x_{t-1}^{(i-1)})} \right)$ <p>accept $x_t^{(i)} = x_t^*$ with probability α, else set $x_t^{(i)} = x_t^{(i-1)}$</p> </div> <p>end for</p> <p>keep the N_p last samples after a burn-in period $\{x_t^{(i)}\}_{i=1}^{N_p} = \{x_t^{(i)}\}_{i=N_b+1}^{N_b+N_p}$</p>

Table 3.8: Joint sequential MCMC algorithm.

In this thesis we use the MCMC algorithm proposed by Septier et al. [95], targeting the joint posterior distribution, that fits better to the visual tracking context, and considerably reduces the computational cost of the MH rule.

3.3.3 Choice of the proposal

The choice of the proposal is one of the most crucial criteria behind the success of MCMC algorithms. The proposal should explore efficiently the state space with reasonable movements. Metropolis and his co-authors have pointed out this issue in their founding paper [67]:

“the maximum displacement must be chosen with some care; if too large, most moves will be forbidden, and if too small, the configuration will not change enough. In either case it will then take longer to come to equilibrium”

A good proposal increases the acceptance probability of the candidate samples and decreases the mixing time of the Markov chain, which accelerates the convergence. If the proposal is not well designed, most candidates are rejected causing the Markov chain to remain in the same state most of the time, which slows down the convergence. In this case, the successive samples can be highly correlated. We can point out here the similarity between the weight degeneracy issue and the mixing properties of the constructed Markov chain.

Several optimality criteria have been proposed to compare Markov chains. They characterize the convergence speed, the variance of the MCMC estimator or the mixing properties [88, 89, 90]. The most commonly used criterion is the expected squared jumping distance: $\text{ESJD} = \mathbb{E} [\|x^{(i+1)} - x^{(i)}\|^2] = 2(1 - \rho_1) \cdot \text{Var}[x^{(i)}]$ where ρ_1 is the first order autocorrelation of the Markov chain. So maximizing the ESJD is equivalent to minimizing the first order autocorrelation. Other criteria can be used, such as the integrated autocorrelation time, the spectral gaps, the mixing time... Despite their plurality, they are quite equivalent [91].

The optimal proposal is directly the target distribution. In this case, the acceptance rate is 1. All the moves are accepted, the successive samples of the Markov chain are independent and the convergence is the fastest. However using MCMC methods assumes that it is impossible to directly sample from the target density.

3. Sequential Monte Carlo methods

The most commonly used proposal in MCMC algorithms is the symmetric random walk . There is a crucial need to improve and adapt this form of proposal to obtain a correct approximation of the posterior density. For that purpose the development of adaptive sampling schemes constitutes a significant improvement of MCMC algorithms.

3.3.4 Adaptive MCMC

In order to enhance the efficiency and the convergence of MCMC algorithms, one should construct Markov chains with good mixing properties. For that purpose, adaptive MCMC algorithms attempt to automatically scale the proposal to approach the optimal proposal. More precisely, they try to find the best parameters for a specific family of proposals.

Here we only consider the most common choice which is a symmetric random walk Metropolis algorithm with a Gaussian proposal $\mathcal{N}(x^{(i-1)}, \sigma^2 I_d)$. The effort is to tune its parameter σ to optimise the MCMC algorithm.

The optimisation is based on a major result showed by Roberts et al. (1997)[86]. They prove that for a random walk Metropolis and for a target density of dimension d written as the product of one dimensional densities, as $d \rightarrow +\infty$, the optimal acceptance rate is equal to 23.4%. In practice this result holds as long as the state space dimension is higher than 4. In the case $d = 1$, the optimal acceptance rate is approximately 44%. This result gives an idea on how well the random walk exploration is.

The key idea of adaptive MCMC is then to learn on the fly the parameter value to reach the optimal acceptance rate. The only constraint is that the adaptation scheme must preserve the stationary distribution of the Markov chain. Indeed a fundamental question when performing adaptation is whether or not adaptation threatens convergence, several authors have addressed this question [35, 1] leading to various conditions that ensure convergence. A lot of adaptive MCMC algorithms have been proposed in the literature. The two most popular algorithms include the adaptive Metropolis of Hario et al.(2001)[35] which tries

to learn the target covariance matrix for a multidimensional random walk and a component-wise alternative, the adaptive Metropolis-Within-Gibbs [87], that adapts the variance of each component of the state independently in order to reach the empirical acceptance rate of 44%.

3.4 Chapter Conclusion

In this chapter we introduced the Bayesian estimation task, and its sequential extension. We introduced two families of Monte Carlo methods, which are simulation based techniques, that allow to solve Bayesian inference in case of non Gaussian and non linear problems.

The first family introduced in this chapter is particle filtering. We discussed the weight degeneracy problem that affects all the importance sampling based methods and reviewed the basic solutions, including resampling schemes, the choice of the importance function and the auxiliary PF and the marginal PF, which are alternatives to the SIS algorithm more robust against weight degeneracy.

The second family is based on the MCMC sampling scheme. We have introduced this alternate way to get a Monte Carlo approximation of a probability density, in particular the Metropolis-Hastings algorithm. Also we discussed the choice of the proposal and its impact on the Monte Carlo approximation, and the improvement by using adaptive sampling schemes.

For both methods, we highlighted the importance of performing a good exploration of the state space, and for that purpose, the choice of the proposal function is crucial to guarantee the quality of the Monte Carlo estimation. Chapter 5 will focus on the optimality of the proposal functions.

Before, in the next chapter we will discuss in more detail how SMC methods are implemented in practice to address the visual tracking problem.

Chapter 4

SMC methods applied to visual tracking

4.1 Introduction

SMC methods have been successfully applied in many areas in science and engineering, such as target tracking, computer vision, navigation, finance, pollution monitoring, communications, audio engineering, biology, meteorology, robotics... where non-linear phenomena frequently arise. In this chapter, we are interested in applying these methods to visual tracking. The implementation of SMC algorithms requires to define the following models and distributions:

- the state model x_t , which contains all the parameters that characterise the tracked objects and that must be estimated.
- the dynamic model and the prior density $p(x_t|x_{t-1})$, which describe the evolution of state between two successive images from *a priori* knowledge on the object behaviour.
- the observation model y_t , which identifies the objects of interest, and the likelihood $p(y_t|x_t)$, which measures the matching between candidate regions represented by samples and the observation model.
- the importance function or the proposal $q(x_t|x_{t-1})$, used to propagate the samples in the state space.

In this chapter, we firstly remind the visual tracking problem formulation and review the main models and distributions proposed in the visual tracking literature. We address all items, except the choice of the proposal, although it is crucial in SMC algorithms to efficiently explore the state space. The proposal, and more precisely the optimal proposal and its approximation, are the subjects of Chapter 5. Then we present and justify the models we have selected in our tracking scheme, in particular the soft detection based observations and the associated likelihood.

4.2 Problem formulation

The objective of visual tracking is to estimate a set of parameters related to the objects of interest from a video sequence. This information, such as location, is not directly accessible, but hidden and embedded in the noisy sequence images. More precisely, the information is assumed to be Markovian and the visual tracking problem can be modelled as a hidden Markov model (HMM), as described in Section 3.1.

This estimation problem, which is almost always non-linear, can then be solved in the Bayesian framework by SMC methods which approximate the posterior density by a set of particles, in particle filters, or by a Markov chain in sequential MCMC methods. In the following, we explain how a SMC algorithm can be implemented to address visual tracking problems.

4.2.1 State model

The first stage to implement a SMC algorithm is to define the state vector x_t , which contains all the parameters that must be estimated: the parameters related to the objects of interest and the auxiliary variables specific to the tracking problem modelling. The choice of these parameters depends on the visual tracking application and on the representation chosen for the objects (silhouettes, set of points or kernels). Here we describe the informations commonly integrated in the state model, we can distinguish between several types of information:

4. SMC methods applied to visual tracking

- The information about the object spatial location. The object position is expressed as coordinates in a 2D or 3D reference system. In general the coordinates of the object center are considered, but some other specific points can also be taken into account.
- The information related to the object geometric representation, especially the shape and the size. The shape can be described by one or several kernels, which can be rectangular, elliptic or interpolated by splines. In this case, the kernel size can be explicitly expressed as width, height, radius... [18, 74] or defined as a scaling factor [127, 57, 52] which gives the ratio between the current size and the initial size. The information can also include the kernel orientation [51, 14, 119]. An other way of modeling the object shape is to use contour models, such as splines, or sets of points: arbitrary points, mesh points or points of interest.
- The information related to the object motion model. They mainly include speed [93, 119, 7] and acceleration which are involved in the dynamic autoregressive models.
- The auxiliary variables specific to the tracking problem modelling. These variables can include information about the status of an object [79, 64] : new/gone object, visible/invisible object, object class, dynamic model associated to the object (when several models are available) [127]... In case of multiple object tracking, the number of objects also needs to be estimated [46, 101].
- The information related to the model hyperparameters. The introduction of hyperparameters in the state vector [18, 11] allows the temporal evolution of some model parameters, which are most often assumed to be constant. For example, it is interesting to estimate the covariance matrix of the state

noise which represents the uncertainty area around the previous state instead of defining a prior value.

The information about the object location and geometric representation are the basic information considered in most visual tracking algorithms, they are often associated with motion information. The use of the other parameters responds to more specific situations. All this information can be combined, which improves the modelling of the visual tracking problem, but also increases the dimension of the state space. In practice, a trade-off must be found between the model accuracy and the computational cost.

4.2.2 Dynamic model and prior density

The state model must be associated to a dynamic model which describes the temporal evolution of the components of the state vector. This dynamic model is described by a transition function $p(x_t|x_{t-1})$, also called the prior density, because it assumes *a priori* knowledge on the state evolution (behaviour of the object and evolution of auxiliary parameters).

In the following, we briefly introduce the main standard dynamic models encountered in visual tracking. Two major categories can be distinguished: Markovian and non-Markovian models.

- The Markovian models take into account the previous states to predict the current one. They include the Gaussian random walk around the previous state with a predefined state noise matrix and the constant speed or acceleration model which is a first or second order autoregressive model. Speed and acceleration are additional parameters which must be estimated by the SMC algorithm. The random walk model imposes few constraints to the object motion and is suitable for all scenarios for which little information about the object evolution is available.

- The non-Markovian models predict the current state independently of the previous ones. For example, the uniform distribution is used in the Wang Landau and stochastic approximation Monte-Carlo (SAMC) sampling algorithms [125, 52, 54]. The advantage of such prior densities is to allow any kind of evolution between two frames without any restriction and to deal with unpredictable motion. In return, they propagate the samples without the assumption of motion continuity considered in Markovian models. The uniform model can be combined with a Markovian model to allow the reset of a particle filter when the object is lost.

When the object can behave in different ways, it is also possible to define multiple dynamic models. In most SMC algorithms, as in the bootstrap filter, the prior density is selected as the proposal. In this case, the algorithms are very easy to be implemented and the computational cost is limited. In particle filters, the weights are directly proportional to the likelihood and in sequential MCMC methods, the acceptance probability is reduced to a likelihood ratio for the random walk model.

However in some scenarios, the object can present abrupt motion that can not be predicted or modelled. When the dynamic model is imprecise and does not reflect the real movement of the tracked objects, most of the samples drawn from the prior density are propagated in unlikely areas of the state space and are wasteful. This case requires the use of the optimal proposal, which considers the current observation in addition to the previous state. In this section, we do not address the problem of state space exploration. The optimal proposal and the main suboptimal strategies to approach it are discussed in detail in Chapter 5.

4.2.3 Observation model and likelihood

The last stage to implement SMC methods is to choose the observations y_t and from them to define the likelihood $p(y_t|x_t)$ used to quantify the matching between the object of interest and a candidate region. More precisely, the likelihood is used to calculate the particle weights in particle filters and the acceptance probability in MCMC methods. It is also interesting to note that the likelihood is also

involved in the expression of the optimal proposal. The choice of the observations is essential because if they do not allow a clear distinction between the object and the other parts of images throughout the video, the tracking process will suffer from various defects, such as target loss, artefact apparition, inaccurate estimation... In visual tracking, the raw data, which include all the pixel values of the sequence images, are too large and too complex to be directly considered as the observations. An image pre-processing is needed to extract a more succinct information. We can distinguish between two types of observations used in SMC methods: those based on appearance models and those provided by detection algorithms.

4.2.3.1 Appearance models

In visual tracking, a lot of SMC algorithms use an appearance model defined from image features to characterize the object to be tracked. Then, the likelihood is defined from a similarity measure or a distance between two appearance models : the model defined on a candidate region (a particle or a proposed value for the Markov chain) and the reference model corresponding to the object of interest. To take into account the appearance changes over time, the reference model needs to be updated during the tracking process.

Among all the available features: pixel values (within image patches) [92], texture (wavelets...), gradient (HOG, SIFT...), contour..., colour is the most widely used. The colour observation model is mainly based on RGB or HSV histograms [71, 78]¹. In this case, the likelihood compares the histogram of a candidate region to the reference histogram using a measure of distance. The most common measure is the Bhattacharyya distance, but other distances have also been considered, such as the diffusion distance or the earth mover's distance (EMD) [18]. The colour histograms have the advantage of being simple and invariant to translation and rotation. But they take into account the statistical distribution of colours, not the spatial distribution. Therefore spatio-colorimetric descriptors have been proposed. One simple way to introduce spatial information is to split the considered region into multiple subregions and to compute a histogram for

¹Except when performing tracking in image sequences provided by specific sensors, such as infra-red cameras, radars etc.

each colour and each subregion. This division can be made vertically / horizontally as in [18] or according to cocentric bounding windows as in [4].

4.2.3.2 Detection information

Many SMC algorithms work from an other type of information: the detected elements (points or regions) provided by a detection algorithm. The likelihood is then defined from a distance between a candidate region and the detection information used as a reference. The selected detector depends on the information required by the application and on the object to be tracked. Here we review the main categories of detectors used in visual tracking [120]:

- Point Detectors provide a set of points of interest in the sequence images. By points of interest we designate pixels with the greatest variation of intensity in their neighbourhood. The most commonly used detectors are Harris, KLT and SIFT (Scale Invariant Feature Transform) [62] that gives better performance by extracting the interest points at different scales and resolutions.
- Image segmentation algorithms deliver a partition of each image into visually similar regions. Each region is characterised by a uniform colour, brightness or texture. These methods are interesting when the object of interest is perceptually different from the neighbouring background and large enough to constitute a distinct area. In visual tracking, segmentation is usually performed by mean shift clustering [15], graph cuts and active contours.
- Background subtraction methods detect the moving objects in the observed scene. The detector first builds a background model based on some image features (colour, edges, motion, texture...) from a training sequence. Then it analyses the incoming images to detect the foreground. All the pixels whose features deviate from the background model are labelled as foreground. A lot of approaches have been proposed. The most used models

4. SMC methods applied to visual tracking

are based on the pixel-wise mixture of Gaussians in the RGB colour space proposed in [104] and its improved versions including fusion of several features, modelling at a region level, foreground modelling...[6].

- Supervised learning algorithms detect specific objects in the scene. The detector is first trained with examples representative of the objects of interest. For that purpose a set of discriminant features (colour, gradients, edges, wavelets...) must be associated to each object class. Then the detector produces an inferred function able to distinguish between the different object classes. A wide range of classifiers are available, including support vector machines (SVM), successfully implemented for human detection in [77, 19], Adaboost [29] combining a set of elementary classifiers and used to detect pedestrians in [112], neural networks and more recently the emerging deep neural networks [106].

Whatever the detector used, the tracking performance highly depends on the quality of the detection information. In general, the detection methods compute a score via a discriminant function and then they decide on the presence or the absence of the object by comparing this score with a threshold. Thus the detection results are provided in binary form and can be described as hard detection information. The threshold is selected to obtain a compromise between the false and missed detection probabilities. These methods are not completely reliable and for any errors, the tracking is deteriorated. An alternate approach is to use soft detection information, that is the detection results before thresholding as in [7, 74]. We precise that detection information is available in any visual tracking system because a detector is always required to automatically detect the apparition of an object of interest during the video. It does not require additional calculations.

Finally, the current trend is to combine several types of information and to include several likelihoods in SMC methods. Most often, the observations are assumed to be conditionally independent given the state. The overall likelihood

then results from the multiplication of the likelihoods defined from each information. The overall likelihood can also be written as a mixture of these different likelihoods. The mixture weights can be chosen according to the level of confidence attached to each information.

4.3 Selected models for single object tracking

In our tracking scheme, the object is represented by a rectangular kernel. The state vector is then defined as $x_k = \{c_k, s_k\}$ with $c_k = \{c_k^x, c_k^y\}$ the position of the top left corner and $s_k = \{s_k^x, s_k^y\}$ the size of the rectangle.

To address any type of movement, we consider a dynamic model with little constraint. As in many works [80, 72, 63, 127], we assume that the components of x_t evolve as mutually independent Gaussian random walks: $x_t|x_{t-1} \sim \mathcal{N}(x_{t-1}, \Sigma)$ where $\Sigma = \text{diag}(\sigma_c^2, \sigma_c^2, \sigma_s^2, \sigma_s^2)$ is the covariance matrix which defines the uncertainty region around the previous state. In real scenarios, the object can perform large amplitude changes in position while the size evolves smoothly. Therefore the position variance σ_c^2 is much larger than the size variance σ_s^2 .

The observation model includes the usual colour information and is enriched with soft detection information extracted from each image I_t .

The colour information is expressed as a set of RGB histograms: $y_t^H = \text{hist}(I_t \cdot \mathbb{1}_{R(x_t)})$ with $R(x_t)$ the region defined by x_t . As in [18], the region $R(x_t)$ is divided into multiple subregions to take into account the colour spatial distribution. A histogram, of size N_{bin} , is then computed for each colour and each subregion.

The colour likelihood $L_H = p(y_t^H|x_t)$ is conventionally defined from the Bhattacharyya distance D_B between the candidate histograms y_t^H and the reference histograms H_t^{ref} for the 3 RGB channels and the S subregions of $R(x_t)$:

$$L_H \propto \exp \left(-\lambda \sum_{p=1}^3 \sum_{r=1}^S D_B^2 \left(y_t^H(p, r), H_t^{ref}(p, r) \right) \right) \quad (4.1)$$

where N_b is the number of bins of the histograms and λ is a tuning parameter

4. SMC methods applied to visual tracking

that determines how peaked the likelihood is, and

$$D_B(h_1, h_2) = \left(1 - \sum_{i=1}^{N_{bin}} \sqrt{h_1 h_2} \right)^{\frac{1}{2}}$$

The soft detection information is provided via a background subtraction algorithm [17] able to detect any kind of moving object. The algorithm models both the background and foreground at a region level by an adaptive mixture of Gaussians in a spatio-colorimetric feature space. Then the pixel classification is based on maximum likelihood and provides a binary mask called the hard detection map. Here, we exploit a richer information that is available in the algorithm before classification. This is the probability map (or soft detection map): $y_t^D = [P_{i,j}]$ where $P_{i,j}$ is the probability that the pixel located at the position (i, j) belongs to the foreground. The soft detection information is more reliable as it takes into account the uncertainties on the object location whereas the thresholding can introduce errors in the hard detection information.

This detection likelihood $L_D = P(y_t^D | x_t)$ is defined from the soft detection map $y_t^D = [P_{i,j}]$ as follows:

$$L_D \propto \exp \left(\lambda_1 \cdot \sum_{(i,j) \in R(x_t)} P_{i,j} - \lambda_2 \cdot N(s_t) \right) \quad (4.2)$$

where $N(s_t)$ is the number of pixels inside the region $R(x_t)$ with size s_t , λ_1 and λ_2 are tuning parameters that define the spread of the likelihood.

In this formulation, the calculation of the detection likelihood is reduced to a sum on the candidate region $R(x_t)$. The penalisation term $\lambda_2 \cdot N(s_t)$ avoids promoting larger candidate regions. We can note that this expression is similar to that proposed in [119], except that it contains the probability $P_{i,j}$ instead of a normalized distance from the background.

Both of these information are fused in the conventional way by assuming that they are conditionally independent given the state [80, 26]. Then the overall

likelihood is written as:

$$p(y_t|x_t) = p(y_t^H|x_t) \cdot p(y_t^D|x_t) = L_H \cdot L_D \quad (4.3)$$

4.4 Chapter Conclusion

In this chapter we show how can Monte Carlo methods be implemented to solve the visual tracking task. We moved from theory to practice and gave an overview of state of the art choices to define: a state vector for kernel based representation, observation models that permits to derive a likelihood function, evolution models that models object displacement and parameter changes a priori.

Finally we presented the models that we have selected for this thesis. We proposed to enrich the observation model with a soft detection information. This information reflects probabilities about the object location and is more reliable than the final binary output which rules on the presence or absence of the object of interest. From this information we derived the associated likelihood which is used for particle weighting and will be exploited for state space exploration.

In the next chapters we will focus on our contributions, namely the near optimal particle filters.

Chapter 5

Near Optimal Particle Filters

As discussed in Chapter 3, the efficiency of Monte Carlo methods strongly depends on the choice of the proposal density used to explore the state space. In sequential methods, its purpose is to guide particles in the most likely areas of the state space between two times. If they are propagated in inappropriate regions, the performance of the algorithms deteriorate significantly. In case of PFs, the importance function has a strong impact on the problem of weight degeneracy. With a well chosen importance function, the importance weights associated to all particles have similar values and their variance remains low. In MCMC context, the proposal affects the convergence speed of the algorithms. A good proposal increases the acceptance probability of the candidate samples and decreases the mixing time of the generated Markov chain.

The simplest choice, originally made in the bootstrap filter [34] or the condensation algorithm in visual tracking [42], is the prior transition density related to the dynamic model. The models considered in the literature are either very simple, including Gaussian random walks and autoregressive models with kinematic parameters, such as velocity and acceleration, or very specific, designed for a particular application. If the dynamic model is imprecise and does not reflect the real movement of the tracked objects, most of the samples generated from the prior distribution are wasteful since they are spread in unlikely areas of the state space. In PFs, the importance weights associated to most particles are very close to zero and in MCMC approaches, most candidate samples are rejected causing

the Markov chain to stand still most of the time. The use of the transition density is based on two assumptions: the knowledge of *a priori* information on the object motion and the motion continuity and smoothness. But in visual tracking applications, the object movement can change quickly and sharply, in particular in case of abrupt motion, and the object displacement between two video frames can be very difficult to predict and model. Therefore a more relevant proposal is required as soon as these assumptions do not hold.

The best choice is the optimal proposal which takes into account the current observations to draw the samples. In this chapter, we first recall the notion of optimal proposal in the cases of PFs and sequential MCMC methods. Unfortunately, using these optimal proposals is generally impossible because the analytic form is unavailable or the computational cost is prohibitive. After reviewing the various suboptimal strategies proposed in the literature, we describe our approach to efficiently explore the state space in visual tracking. More precisely, we present the assumptions that allow us to derive a close approximation of the optimal proposals. From these near optimal proposals, we then derive the corresponding tracking algorithms in PF and MCMC frameworks. The resulting near optimal PF, auxiliary PF, marginal PF and sequential MCMC algorithms are described in detail. The efficiency and robustness of the proposed algorithms will be studied in the next chapter.

5.1 Optimal proposals

5.1.1 Optimal importance function for PFs

As mentioned in section 3.2.5, the optimal importance function, in the sense of weight variance minimisation, accounts for both the previous state and the current observation and is written as:

$$p(x_t|x_{t-1}, y_t) = \frac{p(y_t|x_t) \cdot p(x_t|x_{t-1})}{p(y_t|x_{t-1})} \quad (5.1)$$

where the denominator corresponds to the likelihood prediction given by:

$$p(y_t|x_{t-1}) = \int_{x_t} p(y_t|x_t) \cdot p(x_t|x_{t-1}) \cdot dx_t \quad (5.2)$$

This integral term ensures the normalisation of the optimal importance function.

When using the optimal importance function, the importance weights are simply updated as $w_t^{(i)} \propto w_{t-1}^{(i)} \cdot p(y_t|x_{t-1}^{(i)})$. Thus their variance given the simulated trajectory $x_{0:t-1}^{(i)}$ and the observations $y_{1:t}^{(i)}$ is equal to 0. In practice, using the optimal importance function requires the ability to evaluate the analytic expression, in particular to evaluate the integral term over the current state, and the ability to sample from this expression. In general, this is impossible, except for a few special models.

5.1.1.1 Specific cases

The optimal proposal can be analytically evaluated in two specific cases [2]: for a Gaussian state space model with a linear measurement equation (the transition equation can be nonlinear) and for a finite state space model (with a manageable number of possible states).

Firstly, let us consider the following Gaussian model:

$$\begin{aligned} x_t &= f(x_{t-1}) + v_t, v_t \sim \mathcal{N}(0, \Sigma_v) \\ y_t &= Cx_t + w_t, w_t \sim \mathcal{N}(0, \Sigma_w) \end{aligned}$$

where f is a transition function which can be nonlinear, C is an observation matrix, v_t and w_t are independent white Gaussian noises. In this case, the prior density and the likelihood are Gaussian: $p(x_t|x_{t-1}) = \mathcal{N}(x_t; f(x_{t-1}), \Sigma_v)$ and $p(y_t|x_t) = \mathcal{N}(y_t; Cx_t, \Sigma_w)$.

The likelihood prediction (5.2) is also Gaussian [22]:

$$p(y_t|x_{t-1}) = \mathcal{N}(y_t; Cf(x_{t-1}), \Sigma_w + C\Sigma_vC^t) \quad (5.3)$$

Then the optimal importance function (5.1) is also Gaussian and is written as [22]:

$$p(x_t|x_{t-1}, y_t) = \mathcal{N}(x_t; m_t, \Sigma) \quad (5.4)$$

where $\Sigma = (\Sigma_v^{-1} + C^t \Sigma_w^{-1} C)^{-1}$ and $m_t = \Sigma(\Sigma_v^{-1} f(x_{t-1}) + C^t \Sigma_w^{-1} y_t)$.

Secondly, if the state belongs to a finite set, it is possible to evaluate the numerator of (5.1) pointwise for all the possible values of the state, as well as the denominator since the integral in (5.2) becomes a sum. Then there is no problem to sample from this discrete distribution with a finite support. However this pointwise evaluation becomes computationally too expensive when the number of possible states increases.

In a few cases, the importance function can be chosen hybrid between the optimal and the prior importance functions [39]. This is possible when the state model can be divided into two independent parts and when for one part, evaluating the optimal importance function and sampling from it can be performed easily. The other part can then be drawn from the prior density.

5.1.1.2 General case

Unfortunately, for other models such as those used in visual tracking, the analytic evaluation of the importance function is intractable. As described in chapter 4, most visual tracking problems are formally defined by state space models with linear transition equations but highly non linear measurement equations. In our model described in section 4.3, the prior density is Gaussian and the likelihood is composed of two parts. The colour likelihood is built from the distance between RGB histograms: histograms of the candidate region and reference histograms. The detection likelihood is based on the sum of the foreground probabilities within the candidate region. Hence the prior density is simple, but the likelihood is complex, thus the numerator of (5.1), which involves the product of these two distributions, has no analytic form.

Moreover, when the estimation problem can be turned into a discrete one

with a finite support, the pointwise evaluation of the optimal proposal is most often computationally too expensive. In visual tracking applications, the object center position c_t can be considered as a discrete variable (expressed in pixels) on a finite support corresponding to a video image. Similarly the object size s_t can be considered as a discrete variable (also expressed in pixels) within a wide range from zero size to the image size. Hence the pointwise evaluation of the optimal importance function is impossible because it requires a prohibitive computational time to span the whole state space.

5.1.1.3 Suboptimal strategies - state of the art

In the literature, several suboptimal strategies have been proposed to exploit the current observation in the sampling step and approach the optimal importance function. They can be divided into two main categories.

Implicit approaches first propagate the particles using a simple proposal, generally the prior density, and then add a step to guide them in the most likely areas of the state space from the observation.

- **Auxiliary PF:** The auxiliary PF (APF) [81], already described in section 3.2.6, pre-selects particles according to the most recent observation, before propagation. Performance are really improved when the state noise is small, which is not the case in visual tracking due to the uncertainty of the dynamical model.
- **MCMC step:** The use of MCMC algorithms within PFs make particles move through the state space so that their distribution gets closer to the target distribution. This principle has been first proposed in the resample-move algorithm [31, 127], which consists in applying a Markov kernel with the target distribution as the stationary distribution one or more times after the resampling stage. To increase the convergence speed, an interesting sequence of target distributions, called bridging densities, has been

proposed in [33]. Over the MCMC iterations, they gradually introduce the likelihood information until they reach the posterior density, which allows a series of small transitions instead of a single large transition. The problem is the choice of the bridging densities and the extra computational cost.

- **Optimisation step:** These more recent strategies make particles migrate into areas of high likelihood via an optimization step within PFs. Many methods are based on the mean shift procedure [15, 65] that exploits the local gradient of the likelihood. Other methods use heuristics, such as simulated annealing [20], scatter search [76], particle swarm optimization [45]... The drawback is that these approaches can leave the theoretical framework of PFs, since each particle move can potentially alter the posterior density.

Explicit approaches are more direct and aim to build an importance function from the current observation.

- **Kalman and unscented PFs :** A first solution consists in using extensions of Kalman filter within PFs to approximate the optimal importance function by a Gaussian density. At each time and for each particle, an extended Kalman filter computes the mean and the variance of the Gaussian importance function with the new observation. These moments can be estimated by the extended Kalman filter (EKF) [107] via local linearisation based on the first order Taylor series expansion of the nonlinear functions or more accurately by the unscented Kalman filter (UKF) [109] via a set of sample points. With UKF, the estimates are accurate to the second (even third) order of the Taylor series expansion. The limitation of these methods comes from the Gaussian assumption on the proposal and the extra computational cost.
- **Data driven proposals :** These approaches focus on the design of effective importance functions by exploiting the available information on the current

observation. For visual tracking, several image features can be used, such as colour, texture, motion... The so-called data driven or informative proposal is modelled by a mixture between the prior density and a Gaussian mixture centred on specific points obtained by detection. The means of the Gaussian distributions can be a set of detected interest points as in [80] or the centroids of silhouettes provided by an object detector [72, 63]. Several detectors [127] can also be combined to overcome their imperfections. Performance are improved, but the proposal is still limited to the Gaussian mixture model.

- **Likelihood sampling** : To account for the current observation, the basic idea is to directly use the likelihood to propagate particles [28]. The prior density is then only used to calculate the weights. But for a multi-modal likelihood, a large number of particles are wasteful if the posterior density focuses on a small part of the state space. To overcome this problem, the authors of [108] propose a two-stage sampling process. First particles are drawn from the prior density, then new particles are drawn around the previous ones from the likelihood. The main problem is that it is generally impossible to sample from the likelihood.

5.1.2 Optimal proposal for sequential MCMC

In MCMC context, the optimal proposal, in the sense of fastest convergence, is the target distribution itself. For conventional sequential MCMC methods [46], this distribution is the marginal posterior or filtering distribution:

$$p(x_t|y_{1:t}) \propto p(y_t|x_t) \cdot p(x_t|y_{1:t-1}) \quad (5.5)$$

where the marginal posterior prediction is given by:

$$p(x_t|y_{1:t-1}) \propto \int p(x_t|x_{t-1}) \cdot p(x_{t-1}|y_{1:t-1}) \cdot dx_{t-1} \quad (5.6)$$

An other solution proposed in [95] to simplify the calculation of the acceptance ratio is to use the joint posterior distribution as the target distribution:

$$p(x_t, x_{t-1}|y_{1:t}) \propto p(y_t|x_t) \cdot p(x_t|x_{t-1}) \cdot p(x_{t-1}|y_{1:t-1}) \quad (5.7)$$

When the proposal is equal to the target distribution, the acceptance probability (3.15) is always equal to 1, which means that all the moves are accepted. Then the successive samples of the Markov chain are independent and the convergence is the fastest. However, using MCMC methods assumes that sampling directly from the target distribution is impossible.

5.1.2.1 Suboptimal strategies - state of the art

In the litterature, suboptimal proposals have been proposed to help MCMC methods to improve the mixing properties of the Markov chain and to converge faster to the target density. The following types of proposals have been developped to efficiently explore the state space.

- **Adaptive proposals** : Adaptive MCMC methods [87, 91], already described in section 3.3.4, seek to automatically scale the proposal density to optimize the MCMC sampling process. As synthesized by Roberts and Rosenthal, given a family of proposal densities, the basic idea is to learn and adapt the proposal parameters in order to reach an optimal acceptance rate. The aim is to avoid large displacements resulting in a very low acceptance rate and small displacements resulting in a poor exploration of the state space. Adaptation primarily concerns the proposals based on Gaussian random walks.
- **Composite proposals** : This approach consists in successively using several proposals in MCMC algorithms [95, 68]. The first draw uses a joint proposal to simultaneously update all the components of the state as in the conventional Metropolis-Hastings algorithm. Then a refinement step

is added to propose new local moves. This second draw uses conditional proposals in order to propagate each component of the state separately according to the Metropolis-within-Gibbs algorithm. Both steps generally use Gaussian random walk proposals or independent prior proposals [18]. Other additional stages can be incorporated. In [94] the authors propose to add population MCMC based steps, such as genetic moves and simulated annealing. The limitation of these methods comes from the extra computational cost.

- **Data driven proposals** : This strategy aims to build efficient proposals from the available information on the current observation as in PFs. The so-called data driven or informative proposal is defined from a Gaussian mixture centred on specific points which can be obtained by a detection or a mean-shift algorithm. In case of detection, the centers of the Gaussian distributions can be a set of points of interest or the centroids of silhouettes provided by a human detector [119]. The drawback of these proposals is that they highly depend on the quality and reliability of the detection information. If this information is not correct, the candidate samples are propagated in unlikely regions of the state space and are rejected. In case of mean-shift, the centers of the Gaussian distributions are the new object positions estimated by a mean-shift procedure [123, 36] at the cost of additional computations.
- **Gradient based proposals** : These proposals use the gradient information to move the samples towards the high probability areas of the state space. The new candidates can be drawn according to the principle of Langevin diffusion [122, 123] or Hamiltonian dynamics [114, 97]. The proposal used for Langevin diffusion corresponds to a random walk adjusted by a gradient term. Unfortunately, the convergence of the Markov chain thus generated is no longer ensured [122, 123]. To overcome this limitation, the Metropolis adjusted Langevin algorithm (MALA) [97] has been developed. Hamiltonian dynamics involve the evolution of the state and an auxiliary variable,

called the momentum variable, which corresponds to a kinetic energy term [114, 97]. The advantage of Hamiltonian dynamics is to avoid the inefficient random walk behaviour.

5.2 Proposed approach : approximation of the optimal proposals

To efficiently explore the state space, our objective is to find a compromise between the computational complexity and the optimality of the proposals. The idea is to make the pointwise evaluation of the optimal proposals possible with a reasonable number of calculations. Our approach directly relies on a close approximation of the optimal proposals (5.1), (5.5), (5.7) based on an approximation of the likelihood (4.3), which is the most expensive component in terms of computations. For that purpose, we select the information exploited by the optimal proposal and we introduce assumptions that simplify the computational task.

5.2.1 Information exploited for sampling

According to expression (4.3), the overall likelihood $p(y_t|x_t)$ is built from two types of information: colour information and soft detection information and is equal to the product of two likelihoods (4.1) and (4.2). First let's evaluate the computational cost of the two likelihoods.

- **Colour likelihood** $p(y_t^c|c_t, s_t)$: According to expression (4.1), the evaluation, for a candidate region $R(x_t)$ defined by the state $x_t = \{c_t, s_t\}$, first needs to calculate the corresponding histograms for the 3 colour channels and the S subregions of $R(x_t)$. To obtain the S histograms for a colour channel, the total number of pixels to be analysed is equal to $N(s_t)$, the number of pixels inside the region $R(x_t)$ with size s_t . Then all the N_b bin histograms are compared with the reference histograms, which requires the

calculation of $3 \times S$ Bhattacharyya distances, that cost N_b operations each. The whole process involves $3(N(s_t) + N_b.S)$ operations for each evaluation. In practice, with Matlab, evaluating the colour likelihood for all the pixel locations of an image of size 320×240 takes about 180 seconds for candidate regions of size $s_t=35 \times 60$ divided into $S=4$ parts and histograms with $N_b=10$ bins.

- **Soft detection likelihood** $p(y_t^d | c_t, s_t)$: According to expression (4.2), the evaluation, for a candidate region $R(x_t)$, needs to sum the probabilities $P_{i,j}$ of all the pixels within the region. This involves $N(s_t)$ additions. We do not include in the computational cost all the calculations required to obtain these probabilities $P_{i,j}$, because anyway they have to be calculated in any autonomous visual tracking system: detection information are always necessary to automatically detect the presence of objects of interest. To accelerate detection algorithms without loss of performance, the image is often slightly scaled down. For comparison, with Matlab, evaluating the soft detection likelihood for all the pixels of an image of size 320×240 takes about 50 seconds for candidate regions of size $s_t=35 \times 60$.

We can precise that the computational cost of both likelihoods can be reduced in the same proportions by exploiting redundancy properties. Indeed, a candidate region shares a significant part of pixels with the neighbouring regions in the current image and also with the previous and next images. Therefore it is possible to use some advanced computing methods, such as incremental computing, to avoid making twice the same calculation. Some methods have been developed to faster compute histograms [82, 25] and Bhattacharyya distances [24], similar approaches can easily be extended to the calculation of the soft detection likelihood.

With or without these methods, the likelihood based on soft detection information is computationally much less expensive than the usual colour likelihood. Therefore to significantly reduce the computational cost and time, only the soft detection likelihood is taken into account in the optimal proposal expressions.

Soft detection information, which is a reliable information, showing probabilities about the object location, has already been used in PFs for particle weighting [7], but according to our knowledge, it has never been exploited for particle drawing and state space exploration. Nevertheless the overall likelihood including the colour likelihood is still used in the calculation of particle weights in PF algorithms and of acceptance probabilities in MCMC methods.

5.2.2 Simplifying assumptions

As already mentioned in section 4.3, in most visual tracking works, the position and the size of a tracked object evolve independently, and in case of abrupt changes in position, the size still varies smoothly. The same assumptions are made here.

Furthermore the soft detection likelihood is more sensitive to variations in the position c_t of a candidate region $R(x_t)$ than to variations in its size s_t . As a matter of fact, the position represents the location in the image where the evaluation of expression (4.2) is carried out, while the size represents the number of pixels to be accounted for in the expression.

For illustration, Figure 5.1 compares the soft detection likelihoods displayed as maps and evaluated for all the pixel locations within an image for different size values: (a) the ground truth value, (b) a 20% larger value and (c) a 20% smaller value. We can observe that a significant size change has an influence on the likelihood spread, but causes only a slight variation on the location of the likelihood mode. If we compare the locations of the likelihood maximum, we find an euclidean distance of 14 pixels between cases (a) and (b) and of 8 pixels between cases (a) and (c). In real video scenarios, the size shows less variation and the modes of the likelihood are roughly preserved.

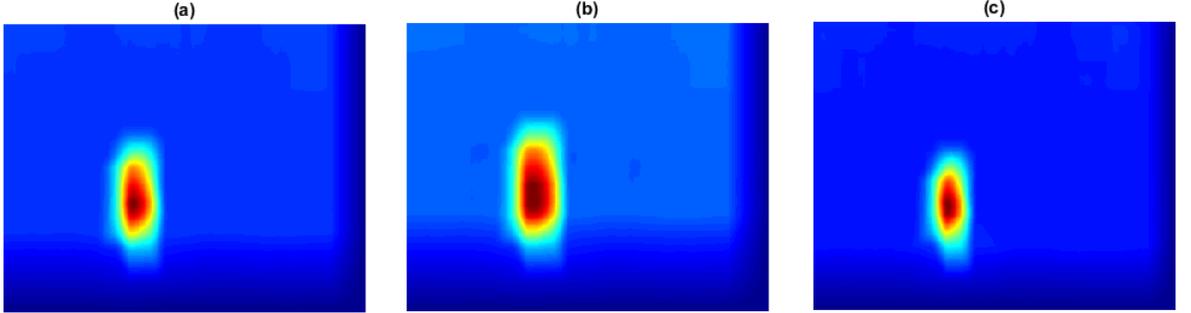


Figure 5.1: Soft detection likelihood for different size values, (a) ground truth value, (b) 20% larger value and (c) 20% smaller value.

For these reasons, the soft detection likelihood is evaluated for a unique value of s_t in the optimal proposal expressions: $\tilde{s}_t = E[s_t | \hat{s}_{t-1}]$ with \hat{s}_{t-1} the estimated size at time $t - 1$. In the case of a Gaussian prior density, $\tilde{s}_t = \hat{s}_{t-1}$.

5.2.3 Approximations of the distributions of interest

From previous hypotheses, we can derive approximations for the distributions involved in the optimal proposal expressions (5.1), (5.5), (5.7). In these expressions, the soft detection observation y_t^d is considered and at each time t , the soft detection likelihood is evaluated for a unique value of the size $\tilde{s}_t = E[s_t | \hat{s}_{t-1}]$. The soft detection likelihood is then approached by:

$$\begin{aligned} \hat{p}(y_t^d | x_t) &= \hat{p}(y_t^d | c_t, s_t) \\ &= p(y_t^d | c_t, \tilde{s}_t) \end{aligned} \quad (5.8)$$

Then the substitution of the likelihood by (5.8) in the expression of the likelihood prediction (5.2) yields the following approximation:

$$\begin{aligned} \hat{p}(y_t^d | x_{t-1}) &= \int \int p(y_t^d | c_t, \tilde{s}_t) \cdot p(c_t | c_{t-1}) \cdot p(s_t | s_{t-1}) \cdot dc_t \cdot ds_t \\ &= \int p(y_t^d | c_t, \tilde{s}_t) \cdot p(c_t | c_{t-1}) \cdot dc_t \end{aligned} \quad (5.9)$$

$$= p(y_t^d | c_{t-1}, \tilde{s}_t) \quad (5.10)$$

By embedding approximations (5.8) and (5.10) in expression (5.1), the optimal importance function can be approached by:

$$\hat{p}(x_t|x_{t-1}, y_t^d) = \frac{p(y_t^d|c_t, \tilde{s}_t) \cdot p(c_t|c_{t-1}) \cdot p(s_t|s_{t-1})}{p(y_t^d|c_{t-1}, \tilde{s}_t)} \quad (5.11)$$

Similarly, we can derive an approximation of the marginal posterior distribution (5.5):

$$\hat{p}(x_t|y_{1:t}) \propto p(y_t^d|c_t, \tilde{s}_t) \cdot p(x_t|y_{1:t-1}) \quad (5.12)$$

and an approximation of the joint posterior distribution (5.7):

$$\hat{p}(x_t, x_{t-1}|y_{1:t}) \propto p(y_t^d|c_t, \tilde{s}_t) \cdot p(c_t|c_{t-1}) \cdot p(s_t|s_{t-1}) \cdot p(x_{t-1}|y_{1:t-1}) \quad (5.13)$$

From these approximations of the optimal proposals, we can now derive the corresponding algorithms in particle filter and sequential MCMC frameworks. We refer to these algorithms as near optimal.

5.3 The near optimal particle filter and its variants

5.3.1 Near optimal particle filter (NOPF)

The near optimal particle filter (NOPF) relies on an implementation of the particle filter using the proposed approximation of the optimal importance function (5.11). To obtain the set of particles at time t from the previous particles $\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_p}$, we first draw the samples $x_t^{(i)}$ using (5.11), which can be rewritten as:

$$\hat{p}(x_t|x_{t-1}^{(i)}, y_t^d) = p(c_t|c_{t-1}^{(i)}, \tilde{s}_t, y_t^d) \cdot p(s_t|s_{t-1}^{(i)}) \quad (5.14)$$

with :

$$p(c_t|c_{t-1}^{(i)}, \tilde{s}_t, y_t^d) = \frac{p(y_t^d|c_t, \tilde{s}_t) \cdot p(c_t|c_{t-1}^{(i)})}{p(y_t^d|c_{t-1}^{(i)}, \tilde{s}_t)} \quad (5.15)$$

The near optimal importance function is written as the product of two densities: a near optimal proposal $p(c_t|c_{t-1}^{(i)}, \tilde{s}_t, y_t^d)$ for the object position c_t and a

prior proposal $p(s_t | s_{t-1}^{(i)})$ for the object size s_t .

The object size is directly drawn from the prior, which is here a Gaussian density according to the dynamic model described in section 4.3. In order to be able to sample the object position, c_t is considered as a discrete variable (expressed in pixels) on a finite support (proportionally scaled to the image size). Then $p(c_t | c_{t-1}^{(i)}, \tilde{s}_t, y_t^d)$ is considered as a discrete distribution which is pointwise evaluated. The denominator of this proposal, which corresponds to a normalizing coefficient, is evaluated by replacing the integral by a sum in expression (5.9):

$$p(y_t^d | c_{t-1}^{(i)}, \tilde{s}_t) = \sum_{c_t} p(y_t^d | c_t, \tilde{s}_t) \cdot p(c_t | c_{t-1}^{(i)}) \quad (5.16)$$

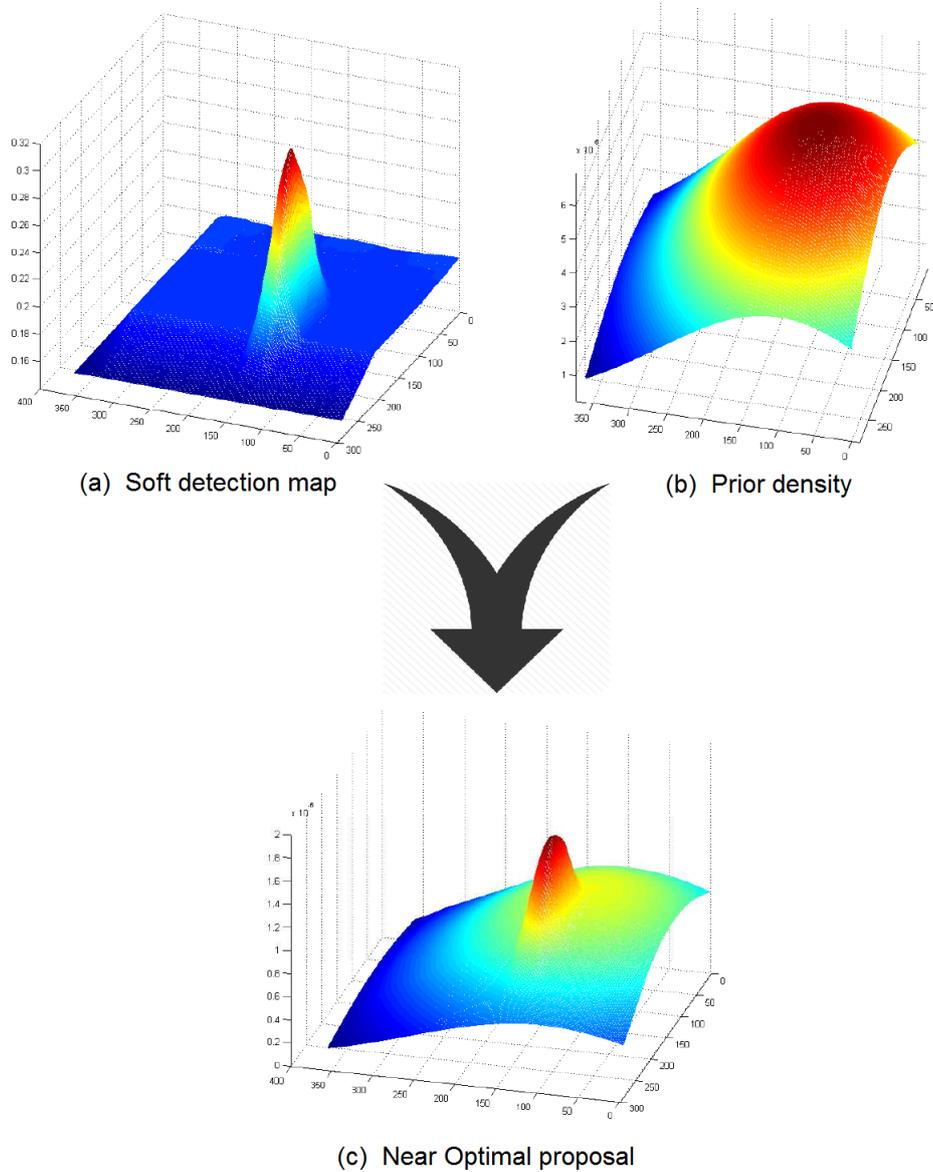


Figure 5.2: Construction of the near optimal proposal

Figure 5.2 shows how the position near optimal importance function is obtained, in the form of a probability map, from both the soft detection likelihood $p(y_t^d | c_t, \tilde{s}_t)$ and the prior density $p(c_t | c_{t-1}^{(i)})$. In the sampling process the position of the particle is drawn from this discrete probability map.

Once the particles are drawn, the importance weights are updated according to (3.4) where the importance function is replaced by the near optimal proposal (5.11). Then we obtain the following recursive expression to update the weights:

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{p(y_t | c_t^{(i)}, s_t^{(i)}) \cdot p(y_t^d | c_{t-1}^{(i)}, \tilde{s}_t)}{p(y_t^d | c_t^{(i)}, \tilde{s}_t)} \quad (5.17)$$

As mentioned previously, the weight calculation involves the overall likelihood including the colour likelihood: $p(y_t | c_t^{(i)}, s_t^{(i)}) = p(y_t^c | c_t^{(i)}, s_t^{(i)}) p(y_t^d | c_t^{(i)}, s_t^{(i)})$.

The algorithm of the NOPF is summarized in Table 5.1.

<p>Initialisation ($t = 0$)</p> <p>sample $\{x_0^{(i)}\}_{i=1}^{N_p} = \{c_0^{(i)}, s_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$</p> <p>initialise $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Sequential processing ($t > 0$)</p> <p>for $i = 1 : N_p$ do</p> <div style="margin-left: 20px;"> <p>sample $s_t^{(i)} \sim p(s_t s_{t-1}^{(i)})$</p> <p>sample $c_t^{(i)} \sim p(c_t c_{t-1}^{(i)}, \tilde{s}_t, y_t^d)$ using (5.15)</p> <p>update $w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{p(y_t c_t^{(i)}, s_t^{(i)}) \cdot p(y_t^d c_{t-1}^{(i)}, \tilde{s}_t)}{p(y_t^d c_t^{(i)}, \tilde{s}_t)}$</p> </div> <p>end for</p> <p>normalise the weights $w_t^{(i)}$</p> <p>estimate $\hat{x}_t = \sum_{i=1}^{N_p} w_t^{(i)} \cdot x_t^{(i)}$</p> <p>if $N_{eff} < \frac{N_p}{3}$</p> <div style="margin-left: 20px;"> <p>resample $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_p}$ to obtain $\{x_t^{(i)}, \frac{1}{N_p}\}_{i=1}^{N_p}$</p> </div> <p>end if</p>

Table 5.1: The NOPF algorithm.

5.3.2 Near optimal auxiliary particle filter (NOAPF)

Now let's incorporate the approximation of the optimal importance function in the auxiliary particle filter (APF) to further improve the state space exploration. The advantage of the APF is to exploit the knowledge about the new observa-

tion to resample particles before their propagation. As previously described in section 3.2.6, the sampling process is performed in two steps. First the particles are resampled according to the first stage weights, which measure the adequacy between the previous particles and the current observation, and then the pre-selected particles are propagated in the state space using the importance function.

More precisely, the proposed approximation allows to derive a near optimal fully adapted auxiliary particle filter (NOAPF). We recall that in the terminology of Pitt and Shephard, an APF is fully adapted when it employs the exact predictive likelihood $p(y_t|x_{t-1}^{(i)})$ to compute the first stage weights and the optimal importance function $p(x_t|x_{t-1}^{(i)}, y_t)$ to propagate the particles. In this case, the weights are all equal.

The NOAPF is obtained by replacing the predictive likelihood and the optimal importance function by their respective approximations $p(y_t^d|c_{t-1}^{(i)}, \tilde{s}_t)$ (5.10) - (5.16) and $p(c_t|c_{t-1}^{(i)}, \tilde{s}_t, y_t^d) \cdot p(s_t|s_{t-1}^{(i)})$ (5.14) in the sampling process. As in the NOPF, the near optimal proposal $p(c_t|c_{t-1}^{(i)}, \tilde{s}_t, y_t^d)$ is considered as a discrete distribution which is pointwise evaluated.

After particle drawing, the importance weights are updated according to (3.12) where the predictive likelihood approximation is replaced by (5.10) and the importance function is replaced by the near optimal importance function (5.14). Then we obtain the following recursive expression to update the weights:

$$w_t^{(i)} \propto \frac{p(y_t|c_t^{(i)}, s_t^{(i)}) \cdot p(c_t^{(i)}|c_{t-1}^{(i)}) \cdot p(s_t^{(i)}|s_{t-1}^{(i)})}{p(y_t^d|c_{t-1}^{(i)}, \tilde{s}_t) \cdot p(c_t^{(i)}|c_{t-1}^{(i)}, \tilde{s}_t, y_t^d) \cdot p(s_t^{(i)}|s_{t-1}^{(i)})} \quad (5.18)$$

By replacing $p(c_t|c_{t-1}^{(i)}, \tilde{s}_t, y_t^d)$ by the expression (5.15), the weight updating expression is simplified as:

$$w_t^{(i)} \propto \frac{p(y_t|c_t^{(i)}, s_t^{(i)})}{p(y_t^d|c_t^{(i)}, \tilde{s}_t)} \quad (5.19)$$

The NOAPF algorithm is summarized in Table 5.2.

<p>Initialisation ($t = 0$)</p> <p>sample $\{x_0^{(i)}\}_{i=1}^{N_p} = \{c_0^{(i)}, s_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$</p> <p>initialise $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Sequential processing ($t > 0$)</p> <p>for $i = 1 : N_p$ do</p> <p style="padding-left: 20px;"> compute the first stage weights $\lambda_t^{(i)} = w_{t-1}^{(i)} \cdot p(y_t^d c_{t-1}^{(i)}, \tilde{s}_t)$ using (5.16)</p> <p>end for</p> <p>normalise the first stage weights $\lambda_t^{(i)}$</p> <p>resample $\{x_{t-1}^{(i)}, \lambda_t^{(i)}\}_{i=1}^{N_p}$ to obtain $\{x_{t-1}^{(i)}, \frac{1}{N_p}\}_{i=1}^{N_p}$</p> <p>for $i = 1 : N_p$ do</p> <p style="padding-left: 20px;"> sample $s_t^{(i)} \sim p(s_t s_{t-1}^{(i)})$</p> <p style="padding-left: 20px;"> sample $c_t^{(i)} \sim p(c_t c_{t-1}^{(i)}, \tilde{s}_t, y_t^d)$ using (5.15)</p> <p style="padding-left: 20px;"> compute $w_t^{(i)} = \frac{p(y_t c_t^{(i)}, s_t^{(i)})}{p(y_t^d c_t^{(i)}, \tilde{s}_t)}$</p> <p>end for</p> <p>normalise the importance weights $w_t^{(i)}$</p> <p>estimate $\hat{x}_t = \sum_{i=1}^{N_p} w_t^{(i)} \cdot x_t^{(i)}$</p>

Table 5.2: The NOAPF algorithm.

5.3.3 Near optimal marginal particle filter (NOMPF)

In this section we derive the near optimal marginal particle filter (NOMPF). As described in section 3.2.7, the interest of the MPF is to perform simulation directly from the marginal posterior distribution $p(x_t | y_{1:t})$ (3.13) to reduce the variance of the importance weights. More precisely, the target distribution is the approximation of $p(x_t | y_{1:t})$ based on the particle estimation $\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_p}$ of the previous marginal $p(x_{t-1} | y_{1:t-1})$:

$$\hat{p}(x_t | y_{1:t}) \propto p(y_t | x_t) \cdot \sum_{i=1}^{N_p} w_{t-1}^{(i)} \cdot p(x_t | x_{t-1}^{(i)}) \quad (5.20)$$

To sample from this distribution, the MPF uses a generic proposal with the

same form as the target distribution : $q(x_t|y_{1:t}) = \sum_{j=1}^{N_p} w_{t-1}^{(j)} \cdot q(x_t|x_{t-1}^{(j)}, y_t)$, with $q(x_t|y_{1:t}) = p(x_t|y_{1:t})$ as the optimal proposal. By only considering the soft detection likelihood and by substituting it by its approximation (5.8) in the previous expression (5.20), we obtain the following near optimal proposal for the MPF :

$$q(x_t|y_{1:t}^d) = p(y_t^d|c_t, \tilde{s}_t) \cdot \sum_{j=1}^{N_p} w_{t-1}^{(j)} \cdot p(c_t|c_{t-1}^{(j)}) \cdot p(s_t|s_{t-1}^{(j)}) \quad (5.21)$$

The problem is that it is impossible to directly sample from this expression. The proposal must be separated into two distinct densities to sample the position and the size independently. Two main strategies can be considered.

5.3.3.1 First algorithm

As we are free to choose any proposal with an appropriate support, we can use a proposal with separate densities to sample the position and the size independently:

$$q_1(x_t|y_{1:t}^d) = p(y_t^d|c_t, \tilde{s}_t) \cdot \sum_{j=1}^{N_p} w_{t-1}^{(j)} p(c_t|c_{t-1}^{(j)}) \cdot \sum_{j=1}^{N_p} w_{t-1}^{(j)} p(s_t|s_{t-1}^{(j)}) \quad (5.22)$$

This proposal is written as the product of two proposal densities: for the object size s_t , a mixture of prior densities $q(s_t|s_{t-1}) = \sum_{j=1}^{N_p} w_{t-1}^{(j)} p(s_t|s_{t-1}^{(j)})$, and for the object position c_t , a near optimal proposal expressed as:

$$q(c_t|c_{t-1}, y_t^d) = p(y_t^d|c_t, \tilde{s}_t) \cdot \sum_{j=1}^{N_p} w_{t-1}^{(j)} p(c_t|c_{t-1}^{(j)}) \quad (5.23)$$

The object size is directly sampled from the mixture of Gaussians. In order to be able to draw the object position, $q(c_t|c_{t-1}, y_t^d)$ is considered as a discrete distribution which is pointwise evaluated.

The importance weight associated to a particle $x_t^{(i)}$ is given by :

$$w_t^{(i)} = \frac{\hat{p}(x_t^{(i)}|y_{1:t})}{q_1(x_t^{(i)}|y_{1:t}^d)} = \frac{p(y_t|c_t^{(i)}, s_t^{(i)}) \cdot \sum_{j=1}^{N_p} w_{t-1}^{(j)} \cdot p(c_t^{(i)}|c_{t-1}^{(j)}) \cdot p(s_t^{(i)}|s_{t-1}^{(j)})}{p(y_t^d|c_t^{(i)}, \tilde{s}_t) \cdot \sum_{j=1}^{N_p} w_{t-1}^{(j)} p(c_t^{(i)}|c_{t-1}^{(j)}) \cdot \sum_{j=1}^{N_p} w_{t-1}^{(j)} p(s_t^{(i)}|s_{t-1}^{(j)})} \quad (5.24)$$

This NOMPF algorithm is given in Table 5.3. The major drawback of this algorithm is the computational cost of the sums, which linearly grows with the number of particles N_p .

<p>Initialisation ($t = 0$)</p> <p>sample $\{x_0^{(i)}\}_{i=1}^{N_p} = \{c_0^{(i)}, s_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$</p> <p>initialise $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Sequential processing ($t > 0$)</p> <p>for $i = 1 : N_p$ do</p> <ul style="list-style-type: none"> draw $s_t^{(i)} \sim \sum_{j=1}^{N_p} w_{t-1}^{(j)} p(s_t s_{t-1}^{(j)})$ draw $c_t^{(i)} \sim q(c_t c_{t-1}, y_t^d)$ using (5.23) update $w_t^{(i)}$ using (5.24) <p>end for</p> <p>normalise the weights $w_t^{(i)}$</p> <p>estimate $\hat{x}_t = \sum_{i=1}^{N_p} w_t^{(i)} \cdot x_t^{(i)}$</p>

Table 5.3: The NOMPF - First algorithm.

5.3.3.2 Second algorithm

The second algorithm is obtained by rewriting the general expression of the near optimal proposal of the MPF (5.21) as :

$$q_2(x_t|y_{1:t}^d) = \sum_{j=1}^{N_p} w_{t-1}^{(j)} \cdot p(y_t^d|c_t, \tilde{s}_t) \cdot p(c_t|c_{t-1}^{(j)}) \cdot p(s_t|s_{t-1}^{(j)}) \quad (5.25)$$

To sample from this mixture, first an index j must be drawn from $\sum_{k=1}^{N_p} w_{t-1}^{(k)} \delta_k$, then $x_t^{(i)} = \{c_t^{(i)}, s_t^{(i)}\}$ must be drawn from the corresponding component of the mixture $p(y_t^d|c_t, \tilde{s}_t) \cdot p(c_t|c_{t-1}^{(j)}) \cdot p(s_t|s_{t-1}^{(j)})$.

5. Near Optimal Particle Filters

The first step is equivalent to directly draw $x_{t-1}^{(j)} = \{c_{t-1}^{(j)}, s_{t-1}^{(j)}\}$ from $\sum_{k=1}^{N_p} w_{t-1}^{(k)} \delta(x_{t-1} - x_{t-1}^{(k)})$, which is also equivalent to resample the set of the N_p particles $\{x_{t-1}^{(k)}, w_{t-1}^{(k)}\}_{k=1}^{N_p}$ and to uniformly draw $x_{t-1}^{(j)}$ from the new set of the particles $\{x_{t-1}^{(k)}, \frac{1}{N_p}\}_{k=1}^{N_p}$.

As in previous algorithms, the object size $s_t^{(i)}$ is easily sampled from the Gaussian prior $p(s_t | s_{t-1}^{(j)})$. In order to be able to draw the object position $c_t^{(i)}$, $p(y_t^d | c_t, \tilde{s}_t) \cdot p(c_t | c_{t-1}^{(j)})$ is considered as a discrete distribution which is pointwise evaluated.

The importance weight associated to a particle $x_t^{(i)}$ is computed by :

$$w_t^{(i)} = \frac{\hat{p}(x_t^{(i)} | y_{1:t})}{q_2(x_t^{(i)} | y_{1:t}^d)} = \frac{p(y_t | c_t^{(i)}, s_t^{(i)}) \cdot \sum_{j=1}^{N_p} w_{t-1}^{(j)} \cdot p(c_t^{(i)} | c_{t-1}^{(j)}) \cdot p(s_t^{(i)} | s_{t-1}^{(j)})}{\sum_{j=1}^{N_p} w_{t-1}^{(j)} \cdot p(y_t^d | c_t^{(i)}, \tilde{s}_t) \cdot p(c_t^{(i)} | c_{t-1}^{(j)}) \cdot p(s_t^{(i)} | s_{t-1}^{(j)})} \quad (5.26)$$

The advantage of this algorithm is that the weight calculation is simplified as :

$$w_t^{(i)} = \frac{p(y_t | c_t^{(i)}, s_t^{(i)})}{p(y_t^d | c_t^{(i)}, \tilde{s}_t)} \quad (5.27)$$

The resulting algorithm is summarised in Table 5.4.

<p>Initialisation ($t = 0$)</p> <p>sample $\{x_0^{(i)}\}_{i=1}^{N_p} = \{c_0^{(i)}, s_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$</p> <p>initialise $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Sequential processing ($t > 0$)</p> <p>for $i = 1 : N_p$ do</p> <ul style="list-style-type: none"> draw $x_{t-1}^{(j)} = \{c_{t-1}^{(j)}, s_{t-1}^{(j)}\} \sim U(\{x_{t-1}^{(i)}\}_{i=1}^{N_p})$ draw $s_t^{(i)} \sim p(s_t s_{t-1}^{(j)})$ draw $c_t^{(i)} \sim p(y_t^d c_t, \tilde{s}_t) \cdot p(c_t c_{t-1}^{(j)})$ compute $w_t^{(i)} = \frac{p(y_t^d c_t^{(i)}, s_t^{(i)})}{p(y_t^d c_t^{(i)}, \tilde{s}_t)}$ <p>end for</p> <p>normalise the weights $w_t^{(i)}$</p> <p>estimate $\hat{x}_t = \sum_{i=1}^{N_p} w_t^{(i)} \cdot x_t^{(i)}$</p> <p>resample $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_p}$ to obtain $\{x_t^{(i)}, \frac{1}{N_p}\}_{i=1}^{N_p}$</p>

Table 5.4: The NOMPF - Second algorithm.

5.4 The near optimal sequential MCMC (NOMCMC)

This section focuses on the near optimal sequential MCMC (NOMCMC). As previously explained, the MCMC method generates a Markov chain that converges to a target density and the optimal proposal is the target density itself. Therefore the NOMCMC relies on an implementation of the sequential MCMC algorithm using the proposed approximation of the marginal or joint posterior distribution (5.12) - (5.13) based on the soft detection information and the simplifying hypotheses.

5.4.1 First algorithm

The first algorithm targets the filtering density $p(x_t | y_{1:t})$, which is approached by the particle estimation of $p(x_{t-1} | y_{1:t-1})$ using the samples $\{x_{t-1}^{(i)}\}_{i=1}^{N_p}$ of the Markov

chain generated at the previous time:

$$\hat{p}(x_t|y_{1:t}) \propto p(y_t|x_t) \cdot \sum_{i=1}^{N_p} \frac{1}{N_p} \cdot p(x_t|x_{t-1}^{(i)}) \quad (5.28)$$

By considering the soft detection likelihood and by substituting it by its approximation (5.8) in the previous expression (5.28), we obtain the following near optimal proposal for the sequential MCMC :

$$q(x_t|y_{1:t}^d) = p(y_t^d|c_t, \tilde{s}_t) \cdot \sum_{j=1}^{N_p} \frac{1}{N_p} \cdot p(c_t|c_{t-1}^{(j)}) \cdot p(s_t|s_{t-1}^{(j)}) \quad (5.29)$$

As in the NOMPF case, the problem is that it is not possible to directly sample from this expression. One solution is to write the proposal as the product of two distinct proposals to sample the position and the size independently. As we are free to choose any proposal with an appropriate support, we can use a proposal with separate densities :

$$q_1(x_t|y_{1:t}^d) = p(y_t^d|c_t, \tilde{s}_t) \cdot \sum_{j=1}^{N_p} \frac{1}{N_p} p(c_t|c_{t-1}^{(j)}) \cdot \sum_{j=1}^{N_p} \frac{1}{N_p} p(s_t|s_{t-1}^{(j)}) \quad (5.30)$$

According to this expression, the object size s_t is directly sampled from a mixture of Gaussian prior densities $q(s_t|s_{t-1}) = \sum_{j=1}^{N_p} \frac{1}{N_p} p(s_t|s_{t-1}^{(j)})$. The object position c_t is drawn from a near optimal proposal expressed as:

$$q(c_t|c_{t-1}, y_t^d) = p(y_t^d|c_t, \tilde{s}_t) \cdot \sum_{j=1}^{N_p} \frac{1}{N_p} p(c_t|c_{t-1}^{(j)}) \quad (5.31)$$

To perform the sampling, $q(c_t|c_{t-1}, y_t^d)$ is considered as a discrete distribution which is pointwise evaluated.

We can note that here the position and the size of a new particle are not derived from the position and the size of a single previous particle, but they come from two different previous particles. In other words, we artificially increase the previous particle set by adding a subset including all the possible combinations

between positions and sizes.

By embedding (5.28) and (5.30) in expression (3.18), we obtain the following acceptance probability:

$$\alpha = \min(1, \gamma) \tag{5.32}$$

with

$$\gamma = \frac{p(y_t | c_t^*, s_t^*) \cdot \sum_{j=1}^{N_p} p(c_t^* | c_{t-1}^{(j)}) \cdot p(s_t^* | s_{t-1}^{(j)})}{p(y_t | c_t^{(i-1)}, s_t^{(i-1)}) \cdot \sum_{j=1}^{N_p} p(c_t^{(i-1)} | c_{t-1}^{(j)}) \cdot p(s_t^{(i-1)} | s_{t-1}^{(j)})} \cdot \frac{p(y_t^d | c_t^{(i-1)}, \tilde{s}_t) \cdot \sum_{j=1}^{N_p} p(c_t^{(i-1)} | c_{t-1}^{(j)}) \cdot \sum_{j=1}^{N_p} p(s_t^{(i-1)} | s_{t-1}^{(j)})}{p(y_t^d | c_t^*, \tilde{s}_t) \cdot \sum_{j=1}^{N_p} p(c_t^* | c_{t-1}^{(j)}) \cdot \sum_{j=1}^{N_p} p(s_t^* | s_{t-1}^{(j)})} \tag{5.33}$$

When computing the acceptance ratio, we unfortunately can not simplify the prior terms as the proposal is not exactly of the same form as the target density. As shown in the expression, the acceptance ratio involves multiple sums over the priors, which is quite costly to compute.

The resulting near optimal MCMC algorithm is described in Table 5.5.

<p>Initialisation ($t = 0$)</p> <p>sample $\{x_0^{(i)}\}_{i=1}^{N_p} = \{c_0^{(i)}, s_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$</p> <p>initialise $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Sequential processing ($t > 0$)</p> <p>initialise the Markov chain $x_t^{(1)} = \{c_t^{(1)}, s_t^{(1)}\} \sim p(x_t)$</p> <p>for $i = 2 : N_b + N_p$ do</p> <ul style="list-style-type: none"> draw $s_t^* \sim \sum_{j=1}^{N_p} \frac{1}{N_p} p(s_t s_{t-1}^{(j)})$ draw $c_t^* \sim q(c_t c_{t-1}, y_t^d)$ using (5.31) compute the acceptance ratio α using (5.32) - (5.33) accept $x_t^{(i)} = x_t^*$ with the probability α, else set $x_t^{(i)} = x_t^{(i-1)}$ <p>end for</p> <p>keep the N_p last samples after a burn-in period of N_b</p> <p>$\{x_t^{(i)}\}_{i=1}^{N_p} = \{x_t^{(i)}\}_{i=N_b+1}^{N_b+N_p}$</p> <p>estimate $\hat{x}_t = \frac{1}{N_p} \cdot \sum_{i=1}^{N_p} x_t^{(i)}$</p>

Table 5.5: The NOMCMC - First algorithm.

5.4.2 Second algorithm

To avoid the computational cost of the sums in the acceptance ratio, the second algorithm targets the joint posterior distribution $p(x_t, x_{t-1} | y_{1:t}) \propto p(y_t | x_t) \cdot p(x_t | x_{t-1}) \cdot p(x_{t-1} | y_{1:t-1})$. This distribution can be approximated by the particle estimation of $p(x_{t-1} | y_{1:t-1})$ based on the samples $\{x_{t-1}^{(i)}\}_{i=1}^{N_p}$ of the Markov chain at the previous time :

$$\hat{p}(x_t, x_{t-1} | y_{1:t}) \propto p(y_t | x_t) \cdot p(x_t | x_{t-1}) \cdot \sum_{i=1}^{N_p} \frac{1}{N_p} \delta(x_{t-1} - x_{t-1}^{(i)}) \quad (5.34)$$

The near optimal proposal can be directly derived from this expression and the previous assumptions:

$$q_2(x_t, x_{t-1} | y_{1:t}^d) = \frac{1}{N_p} \sum_{j=1}^{N_p} p(y_t^d | c_t, \tilde{s}_t) \cdot p(c_t | c_{t-1}^{(j)}) \cdot \delta(c_{t-1} - c_{t-1}^{(j)}) \cdot p(s_t | s_{t-1}^{(j)}) \cdot \delta(s_{t-1} - s_{t-1}^{(j)}) \quad (5.35)$$

5. Near Optimal Particle Filters

To sample from this mixture, first an index j must be uniformly drawn or equally a sample $x_{t-1}^* = \{c_{t-1}^*, s_{t-1}^*\}$ must be drawn from a uniform discrete distribution defined on the set of particles $\{x_{t-1}^{(k)}\}_{k=1}^{N_p}$. Then a sample $x_t^* = \{c_t^*, s_t^*\}$ must be drawn from the corresponding component of the mixture $p(y_t^d | c_t, \tilde{s}_t) \cdot p(c_t | c_{t-1}^*) \cdot p(s_t | s_{t-1}^*)$.

As in previous algorithms, the object size $s_t^{(i)}$ is easily sampled from the Gaussian prior $p(s_t | s_{t-1}^{(j)})$. In order to be able to draw the object position $c_t^{(i)}$, $p(y_t^d | c_t, \tilde{s}_t) \cdot p(c_t | c_{t-1}^{(j)})$ is considered as a discrete distribution which is pointwise evaluated.

The acceptance probability is obtained by integrating (5.34) and (5.35) in expression (3.18). The sums of Dirac delta functions are removed, only the likelihood terms remain. Using this near optimal proposal enables to drastically simplify the acceptance ratio which becomes :

$$\alpha = \min \left(1, \frac{p(y_t | c_t^*, s_t^*) \cdot p(y_t^d | c_t^{(i-1)}, \tilde{s}_t)}{p(y_t | c_t^{(i-1)}, s_t^{(i-1)}) \cdot p(y_t^d | c_t^*, \tilde{s}_t)} \right) \quad (5.36)$$

The resulting near optimal sequential MCMC algorithm is described in Table 5.6.

<p>Initialisation ($t = 0$)</p> <p>sample $\{x_0^{(i)}\}_{i=1}^{N_p} = \{c_0^{(i)}, s_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$</p> <p>initialise $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Sequential processing ($t > 0$)</p> <p>draw $x_{t-1}^* = \{c_{t-1}^*, s_{t-1}^*\} \sim U(\{x_{t-1}^{(i)}\}_{i=1}^{N_p})$</p> <p>initialise the Markov chain $x_t^{(1)} = \{c_t^{(1)}, s_t^{(1)}\} \sim p(x_t)$</p> <p>for $i = 2 : N_b + N_p$ do</p> <ul style="list-style-type: none"> draw $x_{t-1}^* \sim U(\{x_{t-1}^{(i)}\}_{i=1}^{N_p})$ draw $s_t^* \sim p(s_t s_{t-1}^*)$ draw $c_t^* \sim p(y_t^d c_t, \tilde{s}_t) \cdot p(c_t c_{t-1}^*)$. compute the acceptance ratio α using (5.36) accept $x_t^{(i)} = x_t^*$ with the probability α, else set $x_t^{(i)} = x_t^{(i-1)}$ <p>end for</p> <p>keep the N_p last samples after a burn-in period of N_b</p> <p>$\{x_t^{(i)}\}_{i=1}^{N_p} = \{x_t^{(i)}\}_{i=N_b+1}^{N_b+N_p}$</p> <p>estimate $\hat{x}_t = \frac{1}{N_p} \cdot \sum_{i=1}^{N_p} x_t^{(i)}$</p>

Table 5.6: The NOMCMC - Second algorithm.

5.5 Conclusion

This chapter focused on the enhancement of the state space exploration in PF and MCMC frameworks, in particular on the optimal proposals and their approximation. We reviewed the various suboptimal strategies developed in the literature to exploit the current observation in the sampling step and approach the optimal proposals.

Then we proposed a new approach to efficiently explore the state space in visual tracking. Our close approximations of the optimal proposals are based on the soft detection information and are directly derived from the optimal proposals by using simplifying assumptions about the likelihood. Using the soft detection information has two advantages. It is more reliable than the usual binary detection information as it takes into account the uncertainties on the object location and

the corresponding likelihood requires less calculations than the usual colour likelihood defined from a distance between appearance models. Moreover according to our knowledge, this information has never been exploited for particle sampling and state space exploration. In comparison with previous works, the proposed near optimal proposals are no longer limited to the Gaussian model and offer a good compromise between computational complexity and optimality.

From these near optimal proposals, we then derived the corresponding tracking algorithms in PF and MCMC frameworks. The resulting near optimal PF, auxiliary PF, marginal PF and sequential MCMC algorithms were described in detail. For the NOMPF and NOMCMC, two algorithms were proposed. A comparative study, available in Appendix A, shows that in both cases, the second algorithms using the proposals based on mixture densities and the simplest expressions for the acceptance probability achieve the best results. These are the algorithms used in the next chapter dedicated to the efficiency and robustness of the proposed near optimal tracking algorithms.

Chapter 6

Application to abrupt motion tracking

In this chapter we test the efficiency and robustness of the proposed near optimal SMC algorithms including the NOPF (near optimal particle filter), the NOAPF (near optimal auxiliary particle filter), the NOMPF (near optimal marginal particle filter) and the NOMCMC (near optimal Markov chain Monte Carlo). The experiments are realised in the context of abrupt motion. In visual tracking, abrupt motion refers to situations where the object displacement is subject to large uncertainties.

We encounter this phenomenon in basically three types of scenarios:

- **Low frame rate** videos which are common in video surveillance applications using IP cameras. As the amount of data to be transmitted and/or stored is proportional to the number of frames per second (fps), low frame rate videos are privileged for their low transmission and storage cost. The low frame rate can be defined as video streams with a fps lower than the full rate (30 fps). A survey ¹ on the average frame rate used in IP video surveillance shows that more than 70% of users record at 10fps or less, and only 6% higher than 20fps.

¹<http://ipvm.com/reports/recording-frame-rate-whats-actually-being-used>

In our simulations we have downsampled the videos, with a variable DS-rate in order to turn them into low frame rate video streams and simulate abrupt motion.

- **Camera switching** which occurs when using a multi camera system. Indeed when a scene is monitored by several cameras covering different areas, and a target goes out of the field of a camera, we need to switch to another one to keep a track of the target. However as the orientation of these cameras is not the same, the target may appear in an area of the image completely different from where it went out, and in the resulting video stream, we may have an object that instantaneously moved from an image corner to another. This situation brings other issues such as considerable changes of the target appearance and background.
- **Sudden dynamic change** which is more related to the targets behaviour, when this target suddenly changes its dynamics. This specially occurs in security surveillance context and is representative of suspicious behaviours, which make theme of high interest in video surveillance. In this situation the dynamic model used before the behaviour change is no longer capable of performing a successful tracking and the tracker needs adapt quickly otherwise he looses completely the target.

Most tracking methods easily fail to track objects in these complex scenarios. This failure is mainly due to the motion smoothness assumption, made by conventional SMC methods and that does not hold in case of abrupt motion. The proposed near optimal proposals are of high interest to handle the uncertainty of dynamic models encountered in real-world situations. By taking into account the soft detection observations, they are able to capture the motion discontinuity and to guide the samples in the most likely areas of the state space.

This chapter begins with a presentation of the performance metrics used to quantitatively evaluate the tracking algorithms. Then after giving a brief

overview of all the algorithms considered in our experiments, experimental results are shown. First we investigate the general performance of the proposed near optimal SMC methods. Our methods are compared with the conventional SMC methods, the variants for approaching the optimal proposal and with one another. Secondly, we focus on the ability of the near optimal SMC methods to deal with abrupt motion situations and we compare them to the state-of-the-art methods proposed in literature for these situations.

6.1 Performance metrics

In order to quantitatively evaluate the performance of the tracking methods and to compare them, we use the most common performance metrics including the F-measure and the success rate.

F-measure (or F-score) [66]. This metric comes from the information retrieval area. In visual tracking, it is related to the overlapping ratio between the ground truth state and the estimated state. It combines the precision, which corresponds to the proportion of relevant information within the estimated data, and the recall which measures the proportion of the estimated relevant information within all the relevant information:

$$Precision = \frac{E \cap G}{E}$$

$$Recall = \frac{E \cap G}{G}$$

where E is the image region corresponding to the estimated object state and G is the ground truth window.

The F-measure is defined as a trade-off between precision and recall:

$$F - measure = \frac{2 * precision * recall}{precision + recall} \tag{6.1}$$

It may range from 0 when there is no overlap to 1 when the ground truth and estimated regions entirely overlap. The F-measure is able to measure the tracking accuracy in each sequence image. It can also be averaged in order to quantify the performance on the whole sequence by a single value. However the average F-measure is not always relevant, especially when there is much variation among the images of the video.

Success rate. This indicator is preferred to evaluate the global tracking performance for an entire image sequence. The success rate is defined as the ratio between the number of successfully tracked images and the total number of images. There are many ways to decide that an object is correctly tracked [105]. The most widely used rule relies on the F-measure: an object is considered as successfully tracked in an image if the F-measure is larger than 50%.

According to the image sequences, the ground truth can be directly provided in datasets or manually obtained by drawing a bounding box around the object of interest.

6.2 Tracking algorithms

In this section, we present the different SMC methods used for comparison in our experiments and the experimental settings. It seems appropriate to us to compare our near optimal methods to the basic SMC techniques, to the variants for approaching the optimal proposal and to the state-of-the-art methods proposed in literature to handle abrupt motion.

6.2.1 Basic SMC methods

The basic SMC methods are used as a reference for the comparison of the tracking algorithms. The following methods are considered:

- **PF: conventional particle filter** [22]. It is the bootstrap implementation of the sequential importance sampling with the prior function as importance

function and performing a systematic resampling.

- **MCMC: sequential Markov chain Monte Carlo** [46]. This tracker uses the prior density as transition kernel to perform a Gaussian random walk in the state space. In our implementation of the MCMC algorithm we use the framework proposed by Septier et al. [95] that fits better to the visual tracking context, and considerably reduces the computational cost of the MH rule.

6.2.2 SMC methods to approach the optimal proposal

To show the interest of our approach and especially the interest to exploit the soft detection information to propagate the samples, we also consider the main algorithms which aim to approach the optimal proposal by integrating observations in the sampling step or by adapting the proposal on-line:

- **APF: auxiliary particle filter** [81]. This variant of the conventional particle filter performs resampling with knowledge of the current observation. In our implementation of the Auxiliary particle filter we also choose the prior density as proposal distribution and the first stage weights are generated by $E[x_t|x_{t-1}]$, which corresponds to the mean value in the Gaussian model.
- **BPF: boosted particle filter** [72]. It is a sequential importance sampling implementation that relies on the use of a Gaussian mixture as proposal. The first Gaussian is the prior transition model, and the second one is generated from a hard detection information. Indeed, the boosted PF uses a detection results (provided by the Adaboost algorithm described earlier) to detect moving objects in the image. The output of the detector is the centroid of the detected object, this central point is then used as a mean value to generate a Gaussian distribution, with a mixing factor to tune the GMM and give equal or more prominece to one of the components. In our

implementation the mixing factor is fixed at 50%.

- **AMwG: adaptive Metropolis within Gibbs** [91]. This adaptive scheme aims to learn and adapt the parameters of the proposal of each element of the state vector independently. This algorithm computes after each block of proposed candidates the acceptance ratio and then updates its proposal function accordingly, the idea behind this adaptation is to adjust the proposal variance to reach the optimal acceptance ratio that ensures the best results.
- **DDMCMC: data driven MCMC**. The proposal based on hard detection information. As the boosted PF, the DDMCMC has a transition kernel built with a GMM between the prior density and a density centered on the detected objects. Here also we use a mixing factor fixed at 50%.

6.2.3 SMC methods to deal with abrupt motion

To validate the efficiency and the robustness of our near optimal algorithms against abrupt motion, we compare them to the state-of-the-art methods proposed to handle motion uncertainties. These methods include:

- **WLMC: Wang Landau Monte Carlo** [52, 54]. In this method, the sampling scheme uses a Density of state (DoS) grid which corresponds to the partitioning of state space. This DoS grid is used to encourage candidates in low density areas and thus avoid the local maxima trap. This makes the Wang Landau sampling scheme is very interesting when dealing with abrupt motion tracking. In its sampling scheme WLMC uses a non Markovian kernel since it spans uniformly the whole state space.
- **Adaptive WLMC** [54]. Adaptive version of the WLMC algorithm. In this extension of the WLMC, the algorithm selects, after several annealing steps, relevant areas and focuses on. This annealing and subregion selection permits to enhance the precision level of the Monte Carlo estimate.

- **SAMC: Stochastic approximation Monte Carlo and Adaptive SAMC** [125, 126]. This method is similar to the WLMC: Density grid based model, upgrades iteratively the DoS, encourages visiting all subregions and uniform proposal for the object location. The main difference is that the SAMC sampling scheme relies on theoretical background that support its convergence [58].
- **Saliency PF: Saliency based particle filter**[105]. This methods relies on an improved visual saliency model which is integrated to a particle filter to deal with abrupt motion tracking. For that purpose this tracker builds and iteratively adapts a saliency map representing salient regions of the image.
- **HMC: Hamiltonian Monte Carlo** [114]. This method is based on MCMC trackers which integrates Hamiltonian Dynamics to explore the state space. This is realised by introducing a momentum item that permits to construct trajectories according to the Hamiltonian dynamics and thus enhances the random walk exploration.
- **VTD: Visual Tracking Decomposition** [53]. This method is based on the interactive Markov Chain Monte Carlo (IMCMC) framework in which several basic trackers interacts and communicates with one another while running in parallel. For that purpose the observation and motion models are decomposed into multiple basic observation models, each covers a specific appearance of the object, and multiple basic motion models, each one covering a different type of motion. Then each combination of basic observation and evolution models constitutes a basic tracker.
- **LOT: Locally Orderless Tracking**[73]. This algorithm is designed to estimate and adapt, to the rigidity of the tracked object. This flexibility permits to LOT to perform template matching with different constrain levels on the object rigidity. LOT uses a Gaussian-Uniform model where the appearance noise is Gaussian while the localization noise is a mixture-of-uniforms, which permits to cover large areas and address abrupt motion issues.

- **PSO: Particle Swarm optimisation.** PSO is a swarm-based filtering method, in which particles are locally interacting with each other. In this swarm-based sampling strategies, the exploration of the state space is then highly dependent from the social behaviour of the particle swarm. Lim et Al. [59] have enhanced this algorithm by introducing Dynamic Acceleration Parameters (DAP) to cope with the abrupt motion constraints.

6.2.4 General experimental settings

We precise here the settings concerning all the experiments, the settings specific to each test are given in the corresponding paragraphs. For our near optimal methods, we set the model parameters to the following values: $S = 4$ bands to take into account the object colour spatial repartition, $N_b = 10$ bins to calculate the histogram of each RGB channel. The tuning parameters of the likelihood are set empirically to $\lambda = 3$ to set the spread of the colour likelihood, $\lambda_1 = 4, 55 \cdot 10^{-4}$ and $\lambda_2 = 5, 5 \cdot 10^{-5}$ to set the spread of the soft detection likelihood. The noise variance of the object position is chosen large enough to capture abrupt motion: $\sigma_c^2 = 6400$. In case of abrupt motion, the size variation remains smooth so the variance is smaller: $\sigma_s^2 = 2$.

6.3 Performance of near optimal SMC methods

In this section, we investigate the performance of the proposed near optimal SMC methods including the NOPF (near optimal particle filter), the NOAPF (near optimal auxiliary particle filter), the NOMPF (near optimal marginal particle filter) and the NOMCMC (near optimal Markov chain Monte Carlo). Our methods are compared with the conventional SMC methods, the variants for approaching the optimal proposal and with one another. The experiments are realised on real videos extracted from public and own datasets. In order to simulate or/and accentuate abrupt motion, the image sequences are downsampled with different downsampling (DS) rates, as it is done in most papers on abrupt motion tracking.

For a fair comparison, all the algorithms considered in the tests of this sec-

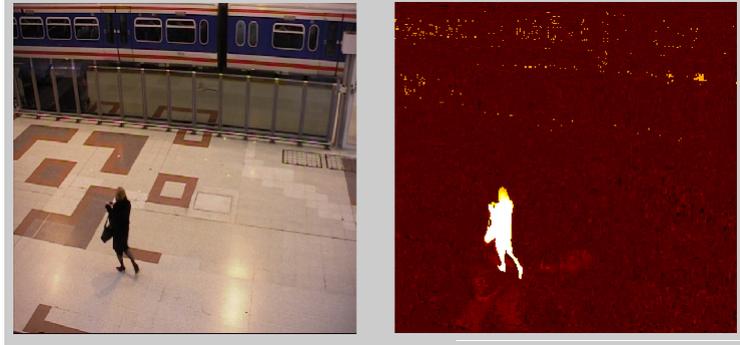


Figure 6.1: Image extracted from the "Walking" sequence with the corresponding soft detection map.

tion have been implemented by ourselves. The main parameters and models are identical. Only the proposal used in the sampling step or/and the sampling mechanism can differ. In particular, the same likelihood, combining colour and soft detection information as described in section 4.3, is used in the weight or acceptance probability calculation stage. And when it is needed, detection information is provided by the same detection algorithm based on background subtraction [17] as described in section 4.3.

As regards simulations, all the tracking results reported in this section have been obtained by an average over 100 realisations. We can also precise that in the first frame of sequences, the tracking is always initialised from the ground truth.

6.3.1 Test Sequences

To conduct the experiments, we have used four image sequences from public and own datasets.

Walking

The "Walking" sequence is taken from the PETS06 public dataset. It contains 101 images of size 720×576 . This sequence constitutes the simplest scenario of the experiment with high contrast between the person and the background, smooth motion and accurate background subtraction.



Figure 6.2: Image extracted from the "Corridor" sequence with the corresponding soft detection map.

Corridor

The "Corridor" sequence comes from our own dataset and consists of 95 images of size 690×540 . This sequence has the following characteristics: background dark areas similar to the person clothes, reflection effect on the metal surface of a lift door, smooth motion. These conditions make the detection information of very poor quality.

Lemming 1

The "Lemming1" sequence is extracted from the public benchmark dataset [116] shared by the visual tracking community. It contains 72 images of size 640×480 . The tracking difficulties come from the complex background with areas similar to the object and the erratic and fast motion of the object. On this sequence, the detection information is noisy.

Lemming 2

The "Lemming2" sequence is also taken from the public benchmark dataset [116]. It includes 82 images of size 640×480 . The tracking difficulties come from the complex background with areas similar to the object and partial occlusions. These conditions make the detection information of very poor quality. The object motion is still erratic, but slower than in the "Lemming1" sequence.

6. Abrupt motion tracking

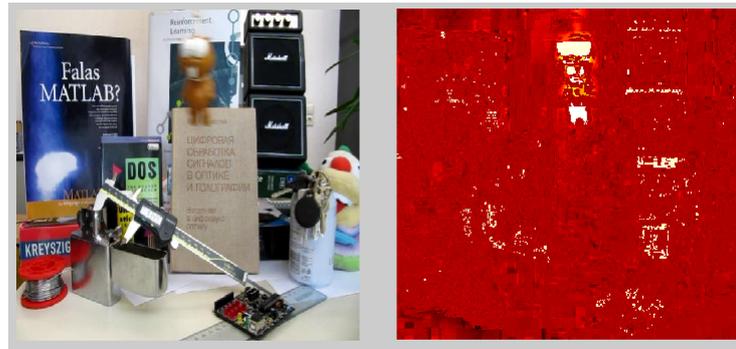


Figure 6.3: Image extracted from the "Lemming 1" sequence with the corresponding soft detection map.



Figure 6.4: Image extracted from the "Lemming 2" sequence with the corresponding soft detection map.

6. Abrupt motion tracking

DS rate	PF	BPF	NOPF
1	80%	83%	84%
10	78%	78%	82%
20	77%	81%	82%
30	76%	82%	83%

Table 6.1: Average F-Measure versus the DS rate for the "Walking" sequence

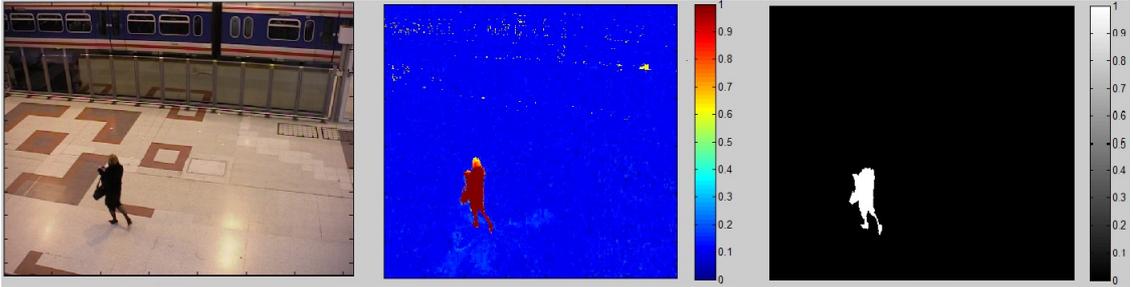


Figure 6.5: Detection maps of the walking sequence (left to right: original image, soft detection map, hard detection map)

6.3.2 Performance of the NOPF

In this experimentation, we evaluate the performance of the NOPF algorithm against abrupt movements. For that purpose we define the following experimental conditions : we use the four sequences described above, the comparison is made between three algorithms the PF, the BPF and our NOPF. For these simulations we use a number of particles $N_p = \{200\}$ for the "Corridor sequence and $N_p = \{100\}$ for "walking", "Lemming 1" and "Lemming 2". Also abrupt movements are emphasized by downsampling the sequences.

Table 6.1 shows the average F-measure versus the DS rate for the "Walking" sequence. This metric is used here instead of the success rate, because this sequence is a simple scenario: high dissimilarity between the target and the background, fixed camera and smooth motion, so the tracking is successful in each frame with the three methods. To compare their accuracy, the average F-measure is needed. The PF has good performance which slightly decrease as the DS rate increases. The BPF and the NOPF perform a little better and their performance are independent of the DS rate.

6. Abrupt motion tracking

The bootstrap PF is able to successfully track the person using a prior distribution with a large position variance, but the accuracy gets lower when the DS rate increases. In this sequence, the hard and soft detection results are of high quality, as shown in figure 6.5. From this detection information, the BPF and NOPF guide the tracker directly towards the most likely areas of the state space and are more efficient than the PF.

DS rate	PF	BPF	NOPF
1	34%	41%	85%
10	27%	30%	71%

Table 6.2: Success rate versus the DS rate for the "Corridor" sequence

Table 6.2 shows the success rate versus the DS rate for the "Corridor" sequence. The PF gives poor tracking results, even for a DS rate of 1. The BPF performs slightly better but still gives unsatisfactory results. The performance obtained with the NOPF are much better, far ahead of the previous methods.

These results are highly related to the complexity of the image sequence characterised by a high similarity between the person and some areas of the background and a reflection effect on the metal surface of a lift door. The PF gets trapped as soon as the person with dark clothes walks by the background dark areas. In these conditions, the detection information is of very poor quality as shown in figure 6.2 and after thresholding, the hard detection information contains errors: missed detection because of appearance similarity and false detection because of reflection effect. Unlike the PF, the BPF retrieves the target after the crossing of the dark areas thanks to the hard detection information, but it loses it quickly after and gets trapped again because of the reflection on the lift door. Although imprecise, the soft detection information is more reliable and allows the NOPF to avoid the local traps.

6. Abrupt motion tracking

DS rate	PF	BPF	NOPF
5	86%	44%	100%
10	62%	42%	97%

Table 6.3: Success rate versus the DS rate for the "Lemming 1" sequence

DS rate	PF	BPF	NOPF
5	86%	69%	100%
10	75%	42%	100%

Table 6.4: Success rate versus the DS rate for the "Lemming 2" sequence

Tables 6.3 and 6.4 respectively represent the success rate for the "Lemming 1" and "Lemming 2" sequences. The performance obtained with the PF depends on the DS rate, it gets lower as the downsampling is accentuated. For these sequences, the BPF performs worse than the PF. Finally the NOPF provides the best performance, whatever the DS rate, with a success rate of 100% or just below.

To understand these results, we point out the difficulties of both sequences: complex background with areas similar to the object, erratic and fast motion in "Lemming 1", partial occlusions in "Lemming 2". Thus the detection information is noisy and of poor quality, which leads to a hard detection information with a high false positive rate. Because the target movement is faster, the detection information are worse for "Lemming 1" than for "Lemming 2". These information used in the BPF tend to propagate particles in unlikely areas of the state space and the tracking performance are considerably smaller than the performance of the conventional PF. As expected, the performance degradation with the BPF is more important for the "Lemming 1" sequence. We can precise that with the PF, the decrease in performance for a DS rate of 10 is also higher for the "Lemming 1" sequence because of the target high speed. Abrupt motion is thus more marked for the same DS rate. The great accuracy of the NOPF again highlights the benefits of using more reliable soft detection information and close approximation of the optimal importance function.

6. Abrupt motion tracking

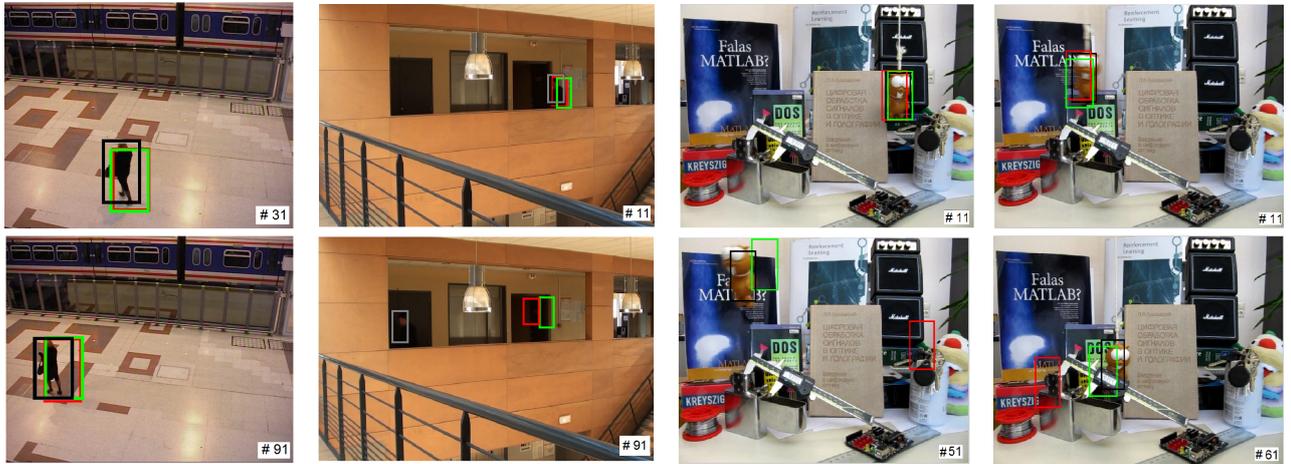


Figure 6.6: Tracking results obtained with PF (green), BPF (red) and NOPF (black) for a DS rate of = 10 (left to right: "Walking", "Corridor", "Lemming 1" and "Lemming 2" sequences).

DS rate	$N_p=10$	$N_p=20$	$N_p=30$	$N_p=50$
5	93,87%	99,33%	99,8%	100%
10	73,25%	84,88%	81,83%	95,63%

Table 6.5: Success rate obtained with the NOPF versus the number of particles for the "Lemming 1" sequence

Now we focus on the impact of the number of particles N_p on the NOPF performance. Table 6.5 shows the success rate obtained with the NOPF on the "Lemming 1" sequence for different values of N_p . In this sequence, the NOPF is able to perform an accurate tracking of the target with a very small number of particles: $N_p = 10$ particles. The performance difference when N_p ranges from 10 to 100 is approximately 7% for a DS rate of 5. These results show that the near optimal importance function used in the NOPF ensures that the drawn particles concentrate in the most likely areas of the state space.

6.3.3 Performance of the NOMCMC

In this experimentation, we evaluate the performances of the NOMCMC algorithm against abrupt movements. For that purpose we define the following experimental conditions : we use three image sequences ("Corridor", "Lemming 1", "Lemming 2"), the comparison is made between five algorithms the MCMC, the AMwG, the DDMCMC, the WLMC and our NOMCMC. For these simulations we use a Number of particles $N_p = \{400\}$ for the "Corridor sequence and $N_p = \{200\}$ for "Lemming 1" and "Lemming 2" and $N_p = \{500, 750\}$ for the WLMC, since the Wang Landau spans the whole image. Also abrupt movements are emphasized by downsampling the sequences.

DS rate	MCMC	AMwG	DD		NOMCMC
			MCMC	WLMC	
1	87.93%	88.15%	40.71%	66.46%	99.9%
5	86.05%	89.38%	41.40%	69.89%	100%
10	77.28%	80%	39.92%	64.53%	99.75%

Table 6.6: Success rate versus the DS rate for the "Lemming 1" sequence

DS rate	MCMC	AMwG	DD		NOMCMC
			MCMC	WLMC	
1	97.33%	97.69%	96.17%	75.86%	100%
5	92.30%	92.71%	87%	78.18%	100%
10	65.54%	59.96%	64.71%	76.18%	97.86%

Table 6.7: Success rate versus the DS rate for the "Lemming 2" sequence

Tables 6.6 and 6.7 show the success rate versus the DS rate for the "Lemming 1" and "Lemming 2" sequences respectively. The MCMC and AMwG perform well for a DS rate of 1, but the performance decrease as the DS rate increases. The online learning of the Gaussian proposal variance improves slightly the tracking accuracy of the AMwG in the "Lemming 1" sequence, there is no improvement in the "Lemming 2" sequence because of partial occlusions. As stated previously for the BPF, the performance of the data driven MCMC depend on the quality

6. Abrupt motion tracking

of the hard detection information, which is poor for the "Lemming 1" sequence and medium for the "Lemming 2" sequence. As a consequence, the DDMCMC is much less efficient than the MCMC for "Lemming 1" and a little less efficient for "Lemming 2".

Unlike the previous methods, the WLMC performance is independent of the DS rate. This is because the WLMC handles each image of the sequence independently without taking into account the time correlation between consecutive images. The method is based on a systematic exploration of all the subregions of the state space at each time. The advantage is that the WLMC can escape local maxima and retrieve a target after losing it, in return without correlation information, it can more easily miss a target. This is highlighted by Figure 6.7 that shows the F-measure per frame obtained by the WLMC and NOMCMC during the "Lemming 2" sequence. The WLMC suffers from multiple and sudden target losses (at frames 8, 23, 27, 31,...) while the NOMCMC is much more stable with a F-measure value around 80%. For both sequences, the NOMCMC clearly outperforms all the other algorithms with a success rate around 100%, even for a DS rate of 10, through better exploration of the state space. The near optimal proposal exploits both the time correlation information via the prior and the soft detection information via the detection likelihood.

DS rate	MCMC	AMwG	DD		NOMCMC
			MCMC	WLMC	
1	50.59%	56.78%	27.92%	32.45%	76.21%
5	42%	42.74%	26.63%	32.24%	70.84%
10	31.29%	34.80%	26.82%	33.17%	65.31%

Table 6.8: Success rate versus the DS rate for the "Corridor" sequence

Table 6.8 represents the success rate for the "Corridor" sequence. As stated before, this sequence is very challenging for the tracking task. The sequential

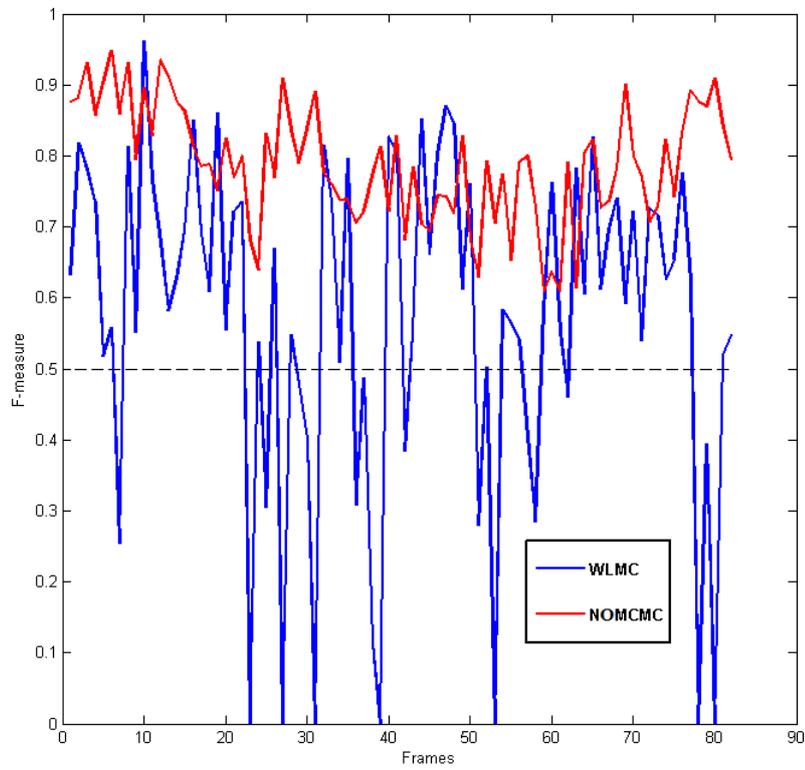


Figure 6.7: F-measure per frame obtained by the WLMC and NOMCMC during the "Lemming 2" sequence.

MCMC has poor tracking performance, even for a DS rate of 1 and the variance adaptation introduced by the AMwG enhances only slightly the results. Both algorithms get trapped in dark regions of the background similar to the walking person. The DDMCMC performs worse, it is misled by the hard detection information which contains errors including missed and false detection because of appearance similarity between the target and the background and reflection effect on the lift door. The performance of the WLMC is a bit better than the DDMCMC performance, but lower than the conventional MCMC and the AMwG. The WLMC gets trapped in the dark regions of the background. Once again, the NOMCMC is far ahead of its competitors with a success rate around 70%. Although noisy, the soft detection information is more reliable and can guide samples in the relevant areas of the state space. Thanks to this information, the NOMCMC avoids the local traps.

6.3.4 Comparison of the different near optimal SMC methods

In this experimentation, we compare the performances of our Near Optimal algorithms in scenarios of abrupt movements. For that purpose we define the following experimental conditions : we use three image sequences ("Corridor", "Lemming 1", "Lemming 2"), the comparison is made between our four Near Optimal algorithms the NOPF, the NOMCMC, the NOMPf and NOAPF. Number of particles $N_p = \{200\}$ for the "Corridor sequence and $N_p = \{100\}$ for "Lemming 1" and "Lemming 2". Also abrupt movements are emphasized by downsampling the sequences.

6. Abrupt motion tracking

Sequence (DS rate)	NOPF	Fully Adapted NOAPF	NOMCMC	NOMPf
LM1 (5)	86.92%	87.08%	85.26%	86.96%
LM1 (10)	84.15%	84.1%	81.13%	85.1%
LM2 (5)	81.11%	81.16%	80.5%	79.9%
LM2 (10)	81.53%	81.28%	80.92%	81%
Corridor (5)	65.68%	66.24%	57.04%	58.54%
Corridor (10)	62.11%	62.5%	42.21%	45.1%

Table 6.9: Average F-measure obtained by the near optimal methods for different sequences and DS rates.

Sequence (DS rate)	NOPF	Fully Adapted NOAPF	NOMCMC	NOMPf
LM1 (5)	100%	100%	100%	100%
LM1 (10)	97.5%	98.25%	97.2%	98%
LM2 (5)	100%	100%	99.9%	99.6%
LM2 (10)	100%	100%	99.7%	100%
Corridor (5)	74.7%	75.11%	65%	67%
Corridor (10)	68%	68.40%	46.6%	52%

Table 6.10: Success rate obtained by the near optimal methods for different sequences and DS rates.

Tables 6.9 and 6.10 respectively summarise the average F-measure and the success rate obtained by the near optimal SMC methods for different image sequences and DS rates. For the "Lemming 1" (LM1) and "Lemming 2" (LM2) sequences, the different implementations of the near optimal approach have approximately the same Success rate around 100%, with very slight differences. For the "Corridor" sequence, which is more difficult, we can observe that the NOPF and NOAPF are more efficient than the NOMCMC and the NOMPf. More precisely, the performance of the NOAPF are slightly above the perfor-

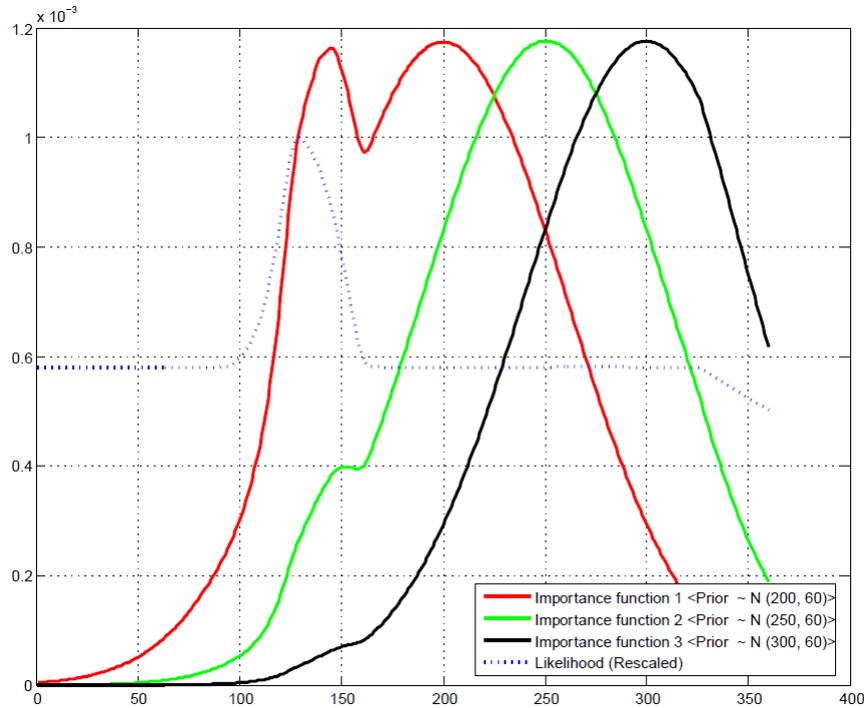


Figure 6.8: Near optimal proposals obtained with different Gaussian prior densities (different means, same variance).

mance of the NOPF thanks to the additional particle preselection step. And the NOMCMC performs a little worse than the NOMPF because it needs more samples to converge. The MCMC is known to outperform particle filters when dealing with high dimension problems, which is not the case of single target tracking.

6.3.4.1 Benefits of particle preselection in NOAPF

As illustrated in figure 5.2, the near optimal importance function is proportional to the product between the soft detection likelihood and the prior density. Let us look at the shape of this product.

Figure 6.8 represents three importance functions obtained by the multiplication of the same soft detection likelihood (in dashed blue) with different Gaussian prior densities. These Gaussian densities have the same variance but different mean values. In SMC methods, these mean values represent the location of par-

ticles in the state space at the previous time. In the first case (in red) when the mean value is close to the likelihood peak, the importance function presents a mode in the area highlighted by the soft detection information and then draws candidates mostly in this area, which is the aim of our approach. If the distance between the peak of the detection likelihood and the prior is large, the importance function inefficiently spreads the candidates on a large portion of the state space with a low probability of reaching the relevant area.

By performing particle pre-selection before the importance sampling step, the APF enables to get closer to the first case. Indeed the resampling is based on the first stage weights, which are computed according to equation (5.16). As these weights are larger when the peak of the detection likelihood and the prior are close, the resampling duplicates the particles which are close to the detection likelihood peak and discards those which are far. Hence the importance sampling step is applied on particles which are the most suitable to be propagated.

Finally, in the NOAPF, we use the detection likelihood twice: first to select the "best" particles, and then to propagate these particles in the state space. This process improves the efficiency of the sampling scheme, as attested by the tracking results in Tables 6.9 and 6.10.

6.3.4.2 Computational time

Figure 6.9 represents the average running time expressed in frames per second (FPS) of our near optimal trackers. The NOPF is the fastest algorithm with a running time of 0.72 FPS. The NOAPF and the NOMPF have slightly lower running times with respectively 0.61 and 0.58 FPS, while the NOMCMC, which includes a burn-in stage, is slower with 0.33 FPS. These results have been obtained by running Matlab codes on a PC with Intel Xeon CPU (3.30 GHz) with 8Go of RAM. Note that the source codes are not optimized in terms of computational efficiency and there are many ways to enhance the speed of the algorithms, especially by speeding up the calculation of the likelihoods and by implementing codes in C/C++ language.

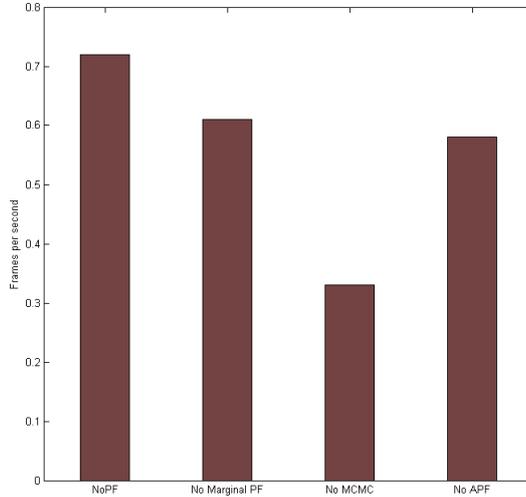


Figure 6.9: Running time of the near optimal trackers expressed in frames per second (FPS).

These results are conform to standard PF based trackers, whose running time generally ranges from 0.5 to 5 FPS [117], and to state-of-the-art algorithms among the highest ranks in the MOT challenge 2016 [70], which run at 0.5 – 0.7 FPS [69]. The competing methods to handle abrupt motion, WLMC and SAMC, have respectively a running time of 2.4 and 2 FPS [124] using C/C++ codes. Finally, it is interesting to precise that a PF using the optimal proposal takes several days per frame, after some simplifications to avoid memory problems. Thus the assumptions made to derive the near optimal proposals significantly speed up calculations while improving the tracking performance.

6.4 Comparison with the state-of-the-art methods for abrupt motion tracking

Now, we focus on the efficiency and robustness of the near optimal PFs and MCMC against motion uncertainties. To this end they are compared with the state-of-the-art methods proposed in the visual tracking literature for abrupt motion tracking. The performance comparison is performed on image sequences

extracted from the abrupt motion tracking benchmark.

The tracking results reported for our near optimal algorithms have been obtained by an average over 100 realisations, each time the tracking initialisation in the first image is made from the ground truth. For all other methods, the tracking performance have been extracted from the journal papers published by the authors. This ensures that the algorithms are set in the most effective way by the authors themselves to capture the abrupt motion in the test sequences.

6.4.1 Test sequences

For this experiment, we have selected 3 videos from public datasets that are commonly used in the visual tracking community to test algorithms against abrupt motion. They correspond to different types of abrupt motion.

Tennis

The "Tennis" sequence contains 31 images of size 512×336 . It has gone through a downsampling with a high DS rate of 25 and thus abrupt motion in this sequence is due to low frame rate. The main characteristics are fast motion with unpredictable directions, pose changes of the tennis player, scale variations when he moves back and forth, camera focus changes, motion and appearance similarity of the other player in the background. In these conditions, the detection information is very noisy.

Ping Pong

The "Ping Pong" sequence consists of 30 images of size 352×240 . The abrupt motion comes from the fast motion of the ball, which is the object of interest, and the sudden direction changes when the ball hits the bat and comes down. The tracking difficulties also come from the small size of the ball and the motion of the player arm and racket in the background, which makes the detection information noisy.

Youngki

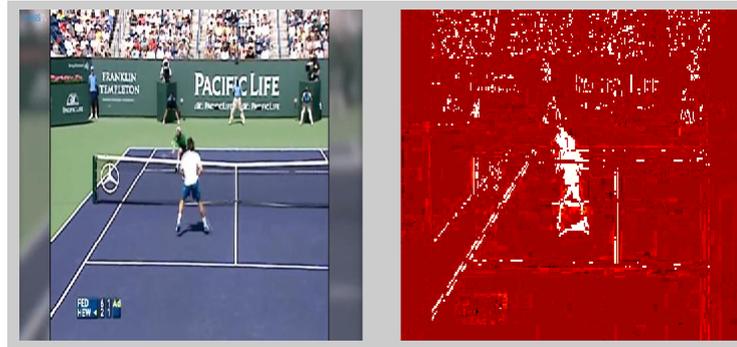


Figure 6.10: Image extracted from the "Tennis" sequence with the corresponding soft detection map.



Figure 6.11: Image extracted from the "Ping Pong" sequence with the corresponding soft detection map.



Figure 6.12: Image extracted from the "Youngki" sequence with the corresponding soft detection map.

The "Youngki" sequence contains 770 images of size 360×240 . This sequence is very challenging because the walking person has a smooth motion, but abrupt motion is caused by camera switching. This abrupt motion is thus characterised by important and sudden changes in the person location, pose and/or scale, and also in the background between two successive frames. Because of background changes and camera shaking, the detection information is very noisy.

6.4.2 Performance comparison

In this experimentation, we compare the performances of our Near Optimal algorithms with state of the art algorithms in scenarios of abrupt movements. For that purpose we define the following experimental conditions : we use the three image sequences ("Tennis", "Ping Pong", "YoungKi"), the comparison is made between 15 algorithms: the NOPF, the NOMCMC, the NOMPf and NOAPF on one side and PF, MCMC, AMwG, WLMC, Adaptive WLMC, Adaptive SAMC, Saliency PF, HMC, VTD, LOT, PSO on the other side. Our Near Optimal algorithms uses $N_p = \{100\}$, for other methods the number of particles vary from 100 to 1000 depending on the method. Also, the result were obtained from several works on abrupt motion tracking [126, 124, 105, 113, 54, 59].

6. Abrupt motion tracking

Methods	Tennis	Stennis (ping pong)	Youngki
Near Optimal PF	78.10%	99.9%	89.02%
Near Optimal MCMC	72.9%	99.67%	62.86%
Near Optimal Marginal PF	74.68%	99.9%	88.24%
Near Optimal APF	76.29%	99.63%	90.34%
PF	35%	42.31%	49.09%
MCMC	43%	46.15%	50.91%
Adaptive MCMC	54%	61.54%	47.01%
WLMG	69%	63.46%	87.27%
Adaptive WLMG	78%	-	86.8%
Adaptive SAMG	84.31%	59.62%	89.87%

Table 6.11: Success rate

6. Abrupt motion tracking

Methods	Tennis	Stennis (ping pong)	Youngki
Near Optimal PF	78.10%	99.9%	89.02%
Near Optimal MCMC	72.9%	99.67%	62.86%
Near Optimal Marginal PF	74.68%	99.9%	88.24%
Near Optimal APF	76.29%	99.63%	90.34%
Saliency PF	62.75%	55.77%	75.58%
HMC	88.24%	67.31%	87.01%
VTD	37.25%	78.85%	65.45%
LOT	43.14%	57.69%	43.18%
PSO	87.3%	87.1%	87.1%

Table 6.12: Success rate

Tables 6.11 and 6.12 compare the success rate obtained by our near optimal SMC methods and the state-of-the-art methods proposed in literature to handle abrupt motion. In these tables, the best performance is highlighted in red and the second highest score is presented in blue. Our algorithms provide the best performance, close respectively to 100% and 90%, for the "Ping Pong" and "Youngki" sequences. For the "Tennis" sequence, the tracking results are not the best, but they remain very good and among the highest, with a success rate around 75%.

Whatever the sequences, the PF, MCMC and adaptive MCMC using the Gaussian prior as proposal give the worst performance. They rely on the motion continuity assumption so they can easily lose the object of interest in case of abrupt motion. Once the object is lost, these algorithms can hardly retrieve it, because they tend to keep searching for the object in the wrong area, leading to error propagation. The online adjustment of the prior variance by the AMCMC helps a little, but it is not enough to capture abrupt motion.

For the "Tennis" sequence, the success rate obtained with the proposed near optimal methods ranges from 72.9% for the NOMCMC to 78.1% for the NOPF. Some qualitative tracking results of these methods are shown in figure 6.13. These results are due to the noisy soft detection information exploited in the near optimal proposal. The camera focus changes quickly, which affects the background modelling and the soft detection quality. Moreover the soft detection information also reflects the motion of the tennis partner and can mislead our trackers when the player comes near the net (as in images #21 and #29 of the figure 6.13). However the tracking performance of our algorithms are still satisfactory.

The VTD and LOT trackers fail to track the abrupt motion in this sequence because the player motion is too unpredictable to be modelled. The saliency PF is slightly better but the saliency information is of poor quality in this sequence. The WLMC and its adaptive version perform better, the sampling scheme based on the exploration of subregions of the state space helps to track abrupt motion, but these algorithms are not stable and can lose the target in some images. One difficulty for all these algorithms is also the appearance similarity between the target and the spectators.

6. Abrupt motion tracking

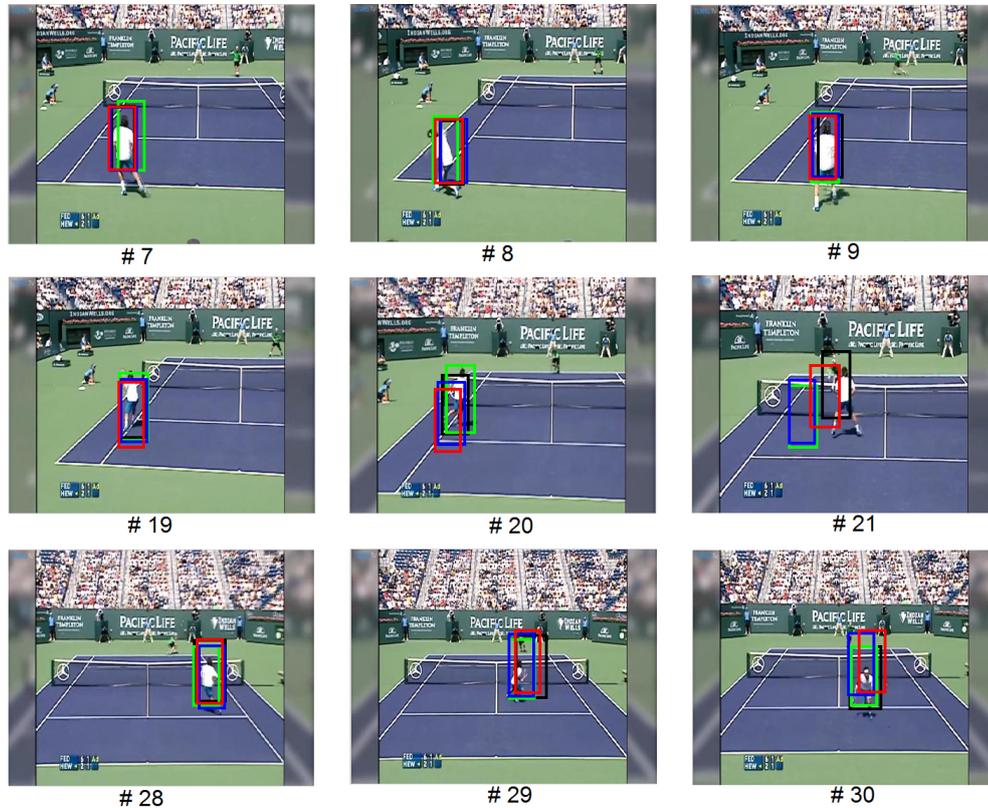


Figure 6.13: Tracking results obtained with NOPF (black), NOMCMC (green), NOMPf (blue) and NOAPF (red) in the "Tennis" sequence.

Only the adaptive SAMC, the HMC and the PSO outperform our near optimal methods.

For the "Ping Pong" sequence, tables 6.11 and 6.12 show that our near optimal trackers have excellent tracking results. With a success rate close to 100%, they are far ahead of the competing methods. As shown in figure 6.14 all of them carry out an accurate tracking of the ball. As the video is shot with a fixed camera, the soft detection is of good quality, except the noise introduced by the hand and the racket movement when hitting the ball. However the near optimal SMC methods do not get trapped by this noise and succeed in accurately capturing the abrupt motion of the ball.

The saliency PF, HMC, WLMC, AWLMC, ASAMC and LOT perform much

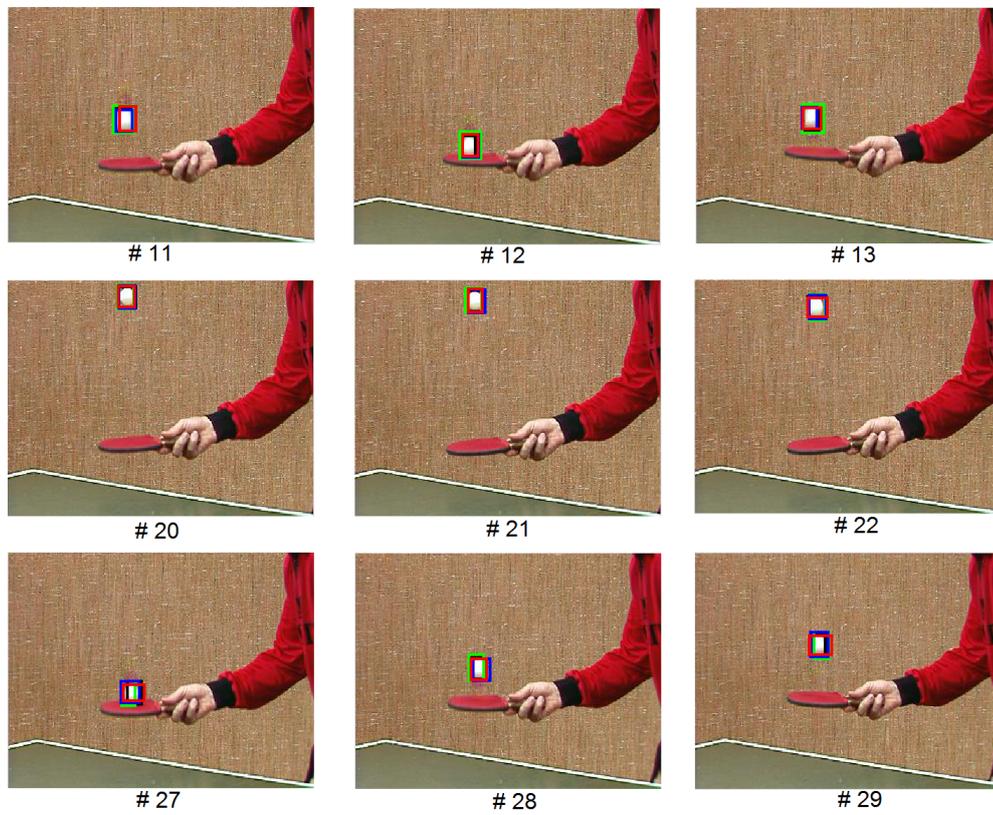


Figure 6.14: Tracking results obtained with NOPF (black), NOMCMC (green), NOMPf (blue) and NOAPF(red) in the "Ping Pong" sequence.

less with a success rate around 60%. On certain images, the saliency information mislead the PF because it highlights not only the ball but also the racket, the player hand and arm. The grid based methods WLMC, AWLMC, ASAMC are known to be less efficient when the resolution of the object of interest is small. In this case, the state space should be divided into small subregions, which significantly increases the running time of the algorithm. All these methods can track the ball on some images, but not continuously. The VTD and PSO perform better with a respective success rate of 78.85% and 87.1%.

For the "Youngki" sequence, the NOPF and the NOAPF provide the best performance with a success rate around 90%. According to figure 6.15, the NOAPF is the fastest algorithm to retrieve the target just after camera switching. The momentary target loss between images #117 and #118 - #119 is explained by the high level of noise in the soft detection information used by the near optimal proposal. For illustration purposes, the soft detection maps obtained on the frames #118 and #397 of the sequence are shown in figure 6.16. On image #397, the soft detection is less noisy and our algorithms are able to catch the important and sudden changes in the target position, pose and scale. The tracking results obtained for this sequence show that the near optimal proposal, which is more precisely the product of a near optimal proposal for the position and a prior proposal for the size, efficiently guides the samples even when the target scale varies more significantly.

At each camera switching, the PF and MCMC fail to escape from the previous location and lose the target. VTD and LOT perform slightly better but still have unsatisfactory results as the motion of the target is difficult to model and its appearance significantly changes over time. The performance of the saliency PF, HMC, PSO, WLMC, AWLMC and ASAMC are better, but slightly worse than the performance of our algorithms. The sampling schemes used by the grid based algorithms can track sudden dynamic changes at each time since they search in the whole state space regardless of the previous state.

Finally all these simulations prove that the proposed near optimal SMC meth-

6. Abrupt motion tracking



Figure 6.15: Tracking results obtained with NOPF (black), NOMCMC (green), NOMPF (blue) and NOAPF (red) in the "Youngki" sequence.

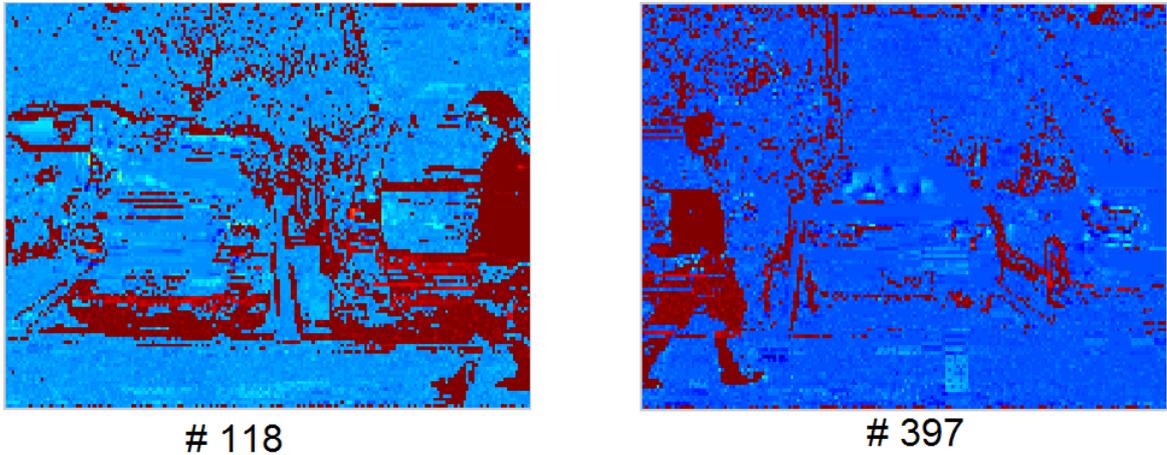


Figure 6.16: Soft detection maps obtained on the frames #118 and #397 of the "Youngki" sequence.

ods are efficient and robust to handle the different types of abrupt motion. Compared to the state-of-the-art methods proposed in literature for abrupt motion tracking, they are competitive and outperform numerous methods.

6.5 Chapter Conclusion

In this chapter we showed the efficiency and the robustness of our Near Optimal algorithms. First we showed the capabilities of the NOPF and NOMCMC to handle abrupt motion in real life scenarios. Moreover we compared between our Near Optimal PFs and MCMC to highlight the interest of particle pre selection and how it enhances the performances of the NOAPF. Finally we compared our methods with other state of the art methods, developed specifically to handle abrupt motion, and proved that our near optimal trackers are very competitive, robust and trustworthy against abrupt motions.

In the next chapters we will focus on other contributions, where we extended the Near Optimal approach to multiple object tracking and introduced the Local PF to reduce the dimensional growth of multiple object tracking.

Chapter 7

Multiple object tracking

Multiple object tracking (MOT) is still a challenging task in visual tracking. In very general terms, the problem of MOT can be solved in two different ways. A common assumption is that each object moves independently of the others. In this case, it is possible to run multiple independent PFs or MCMC algorithms, one for each object. In comparison with single object tracking, the dimension of the state space remains unchanged and tracking performance is similar. However in practice, this assumption is not always correct and this approach is susceptible to tracking failures when the objects interact. Using an interaction model requires to consider a joint configuration space and to run a single joint PF or MCMC algorithm. In this second approach, when the number of tracked objects increases, the state space becomes increasingly large, the number of particles required to find the areas of interest grows exponentially with the state space dimension and the computational cost quickly becomes too expensive.

In this chapter, after an overview of the existing Monte Carlo methods developed for MOT, we extend the near optimal proposals to MOT scenarios to improve the exploration of the state space. By taking into account the soft detection information, these proposals are able to propagate the particles in the regions of interest of the state space and thus to cleverly explore the high-dimensional state space with a limited number of particles. However, it is difficult to adapt the near optimal proposals to MOT models which take into account the interactions of the objects. To improve the tracking performance further while exploiting the

dependencies between the objects, we propose to use the local PF recently developed by Rebeschini et al. The initial idea is that interactions are local: the dynamics and observations related to an object depend only on the neighbouring objects. The local PF allows to combine the interaction modelling and the partition of the large state space into separate subspaces of smaller dimension. Experimental results demonstrate the benefits of the proposed methods.

7.1 Related work

In past years, Bayesian state space formulation and probabilistic methods have gained a great interest and have shown capabilities to address the MOT problem. Among these methods, two main categories can be distinguished.

The first category defines a single object state space and runs in parallel several independent SMC algorithms, basically PFs, one for each object. This simple approach is used in many works [83, 9, 46, 55, 7]. The advantage is that the dimension of the state space remains small. But the major drawback of these methods is that no interaction or dependency can be modelled between the different objects within the PFs and that observations generated by the other targets are considered as noise or false positive. At the expense of an additional cost, a separate processing can be added to deal with the interactions. Using independent trackers also requires solving a data association problem to assign the observations, detections in most cases, to the objects [103]. The most commonly used approaches include the joint probabilistic data association filter (JPDAF) [27, 13] and the multiple hypothesis tracking (MHT) [85].

The second family offers a more rigorous formulation of the problem based on a joint state space configuration. Starting from the initial work of Avitzour et al. [3] who introduced a PF over a state space made from the concatenation of the multiple objects, a lot of joint PFs have been developed [40, 23, 49, 111, 43, 110]. This joint formulation has the advantage of offering an elegant framework where the different objects can be labelled and the interactions can be explicitly taken into account. But it induces a considerable growth of the state space dimension,

which highly increases the complexity of the problem. PFs require a number of particles that increases exponentially with the number of objects, so the computational cost quickly becomes too expensive. This problem is known as the curse of dimensionality [96] and it makes PFs ineffective in high dimensional applications such as MOT.

Therefore alternative methods have been developed to deal with high dimensional problems in visual tracking. Sequential MCMC methods, previously described in Chapter 3, are known to be more effective than PFs in high dimensional state spaces [95]. They have been much used for multiple object visual tracking [119, 36, 123] since the initial work of Khan et al. [46]. An other solution consists in using PFs with partitioned sampling (PS). First introduced in [64], this method decomposes the state space into a partition. The algorithm separately samples each subspace, associated with an object or a group of objects, and performs a resampling routine before moving to the next subspace. However this process leads to an impoverishment and a diversity loss of the particles due to the successive resampling procedures. The order in which the subspaces are explored has a strong impact on the performance [102]. To overcome this limitation, several improvements have been proposed, such as dynamic PS [102] or ranked PS [115].

In this chapter, to explore the state space more effectively and enhance the tracking capabilities of both independent and joint PFs, we propose to adapt the near optimal importance function to MOT. To overcome the dimension problem in the joint configuration, we also propose to use the local particle filter [84], which partitions the large state space into blocks of smaller dimension as PS, but without being affected by the treatment order.

7.2 Near optimal proposal for multi-object tracking

Handling multiple objects in a visual tracking scenario introduces additional complexity. As stated previously, one of the main issues encountered when performing MOT with PFs is the increase of the problem dimension, in case of a joint state

space model. Better proposals are required to propagate the particles in the regions of interest of the state space and thus to explore state space with a limited number of particles. Here we extend the near optimal importance function to both configurations: independent and joint PFs. Since it is difficult to do the extension to MOT models which take into account the interactions between the objects, we consider a model with independent objects.

7.2.1 Problem formulation - Tracking of multiple independent objects

We assume a fixed number of objects. The state vector is simply defined as the concatenation of the individual target states: $X_t = \{x_t^1, x_t^2, \dots, x_t^{N_o}\}$ with N_o the number of objects.

Each object j is represented by a rectangular kernel evolving according to a quasi constant velocity model. Then $x_t^j = \{c_t^j, v_t^j, s_t^j\}$ with $c_t = \{c_t^x, c_t^y\}$ the position of the top left corner, $s_t = \{s_t^x, s_t^y\}$ the size of the rectangle and $v_t = \{v_t^x, v_t^y\}$ the velocity between two successive images.

We assume that there is no interaction between the objects, thus they evolve independently of one another and the prior density is written as the product of the individual prior densities:

$$p(X_t|X_{t-1}) = \prod_{j=1}^{N_o} p(x_t^j|x_{t-1}^j) \quad (7.1)$$

To address a lot of types of movement and to take into account the moving direction of the objects, the quasi constant velocity model is used. The state of each object x_t^j is given by:

$$\begin{cases} c_t^j = c_{t-1}^j + v_{t-1}^j + \epsilon_c \\ v_t^j = v_{t-1}^j + \epsilon_v \\ s_t^j = s_{t-1}^j + \epsilon_s \end{cases} \quad (7.2)$$

where the state noises ϵ_c , ϵ_v , ϵ_s are independent white Gaussian noises with

$\Sigma_c = \text{diag}(\sigma_c^2, \sigma_c^2)$, $\Sigma_v = \text{diag}(\sigma_v^2, \sigma_v^2)$ and $\Sigma_s = \text{diag}(\sigma_s^2, \sigma_s^2)$ the respective covariance matrices defining the uncertainty region around the previous states.

We also assume that the likelihood is the product of the individual object likelihoods:

$$p(y_t|X_t) = \prod_{j=1}^{N_o} p(y_t|x_t^j) \quad (7.3)$$

where $p(y_t|x_t^j)$ combines the colour likelihood $p(y_t^c|x_t^j)$ (4.2) and the detection likelihood $p(y_t^d|x_t^j)$ (4.1) defined in section 4.3. As in the observation model used for single object tracking, both likelihoods only depend on the position and the size of the state x_t^j .

We consider here a fixed number of objects but the approach can be easily extended to deal with a time varying number of objects by using a random finite set or easier by fixing a maximum number of targets and associating to each object an existence variable [46, 95, 36, 123, 119]. The birth and death of each target must also be modelled.

7.2.2 Multiple independent near optimal particle filters

Using multiple independent particle filters requires to define a near optimal importance function per PF or object, which is then similar to the one defined in Chapter 5. As the state vector is enriched with a velocity variable, the near optimal importance function is written (similarly to expression (5.11)) as:

$$\hat{p}(x_t^j|x_{t-1}^j, y_t^d) = \frac{p(y_t^d|c_t^j, \tilde{s}_t^j) \cdot p(c_t^j|c_{t-1}^j, v_{t-1}^j) \cdot p(s_t^j|s_{t-1}^j) \cdot p(v_t^j|v_{t-1}^j)}{p(y_t^d|c_{t-1}^j, v_{t-1}^j, \tilde{s}_t^j)} \quad (7.4)$$

with

$$p(y_t^d|c_{t-1}^j, v_{t-1}^j, \tilde{s}_t^j) = \int p(y_t^d|c_t^j, \tilde{s}_t^j) \cdot p(c_t^j|c_{t-1}^j, v_{t-1}^j) \cdot dc_t^j$$

7. Multiple Object Tracking

As in (5.14), the near optimal proposal can be rewritten as:

$$\hat{p}(x_t^j | x_{t-1}^j, y_t^d) = p(c_t^j | c_{t-1}^j, v_{t-1}^j, \tilde{s}_t^j, y_t^d) \cdot p(s_t^j | s_{t-1}^j) \cdot p(v_t^j | v_{t-1}^j) \quad (7.5)$$

with :

$$p(c_t^j | c_{t-1}^j, v_{t-1}^j, \tilde{s}_t^j, y_t^d) = \frac{p(y_t^d | c_t^j, \tilde{s}_t^j) \cdot p(c_t^j | c_{t-1}^j, v_{t-1}^j)}{p(y_t^d | c_{t-1}^j, v_{t-1}^j, \tilde{s}_t^j)} \quad (7.6)$$

This formulation shows the product of three proposal densities: a near optimal proposal for the object position and prior proposals for the size and the velocity. These two components can be directly sampled from the Gaussian priors. In order to be able to draw the object position, $p(c_t^j | c_{t-1}^j, v_{t-1}^j, \tilde{s}_t^j, y_t^d)$ is considered as a discrete distribution which is pointwise evaluated.

The multiple independent NOPFs are directly obtained by implementing one NOPF per object. The corresponding algorithm is summarised in Table 7.1.

<p>Initialisation ($t = 0$)</p> <p>sample $\{x_0^{j,(i)}\}_{i=1}^{N_p} \sim p(x_0), \forall j = 1, \dots, N_o$</p> <p>initialise $\{w_0^{j,(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}, \forall j = 1, \dots, N_o$</p>
<p>Sequential processing ($t \geq 0$)</p> <p>for $j = 1 : N_o$</p> <div style="border-left: 1px solid black; padding-left: 10px;"> <p>for $i = 1 : N_p$</p> <div style="border-left: 1px solid black; padding-left: 10px;"> <p>sample $s_t^{j,(i)} \sim p(s_t^j s_{t-1}^{j,(i)})$</p> <p>sample $c_t^{j,(i)} \sim p(c_t^j c_{t-1}^{j,(i)}, v_{t-1}^{j,(i)}, \tilde{s}_t^j, y_t^d)$ using (7.6)</p> <p>sample $v_t^{j,(i)} \sim p(v_t^j v_{t-1}^{j,(i)})$</p> <p>update $w_t^{j,(i)} = w_{t-1}^{j,(i)} \cdot \frac{p(y_t^d c_t^{j,(i)}, s_t^{j,(i)}) \cdot p(y_t^d c_{t-1}^{j,(i)}, v_{t-1}^{j,(i)}, \tilde{s}_t^j)}{p(y_t^d c_{t-1}^{j,(i)}, v_{t-1}^{j,(i)}, \tilde{s}_t^j)}$</p> </div> <p>end for</p> <p>normalise the weights $w_t^{j,(i)}$</p> <p>estimate $\hat{x}_t^j = \sum_{i=1}^{N_p} w_t^{j,(i)} \cdot x_t^{j,(i)}$</p> <p>resample $\{x_t^{j,(i)}, w_t^{j,(i)}\}_{i=1}^{N_p}$ to obtain $\{x_t^{j,(i)}, \frac{1}{N_p}\}_{i=1}^{N_p}$</p> </div> <p>end for</p>

Table 7.1: Multiple independent NOPF algorithm.

7.2.3 Joint near optimal particle filter

In the joint state space configuration, the estimation of the state vector X_t is made by using one single particle filter that targets the joint posterior distribution $p(X_t|y_t) = p(x_t^1, x_t^2, \dots, x_t^{N_o}|y_t)$.

From expressions (7.1) and (7.3), the joint near optimal importance function is written as the product of the individual near optimal importance functions:

$$\begin{aligned}
 p(X_t|X_{t-1}, y_t) &= \frac{p(y_t|X_t) \cdot p(X_t|X_{t-1})}{p(y_t|X_{t-1})} \\
 &= \prod_{j=1}^{N_o} \frac{p(y_t|x_t^j) \cdot p(x_t^j|x_{t-1}^j)}{p(y_t|x_{t-1}^j)} \\
 &= \prod_{j=1}^{N_o} p(x_t^j|x_{t-1}^j, y_t)
 \end{aligned} \tag{7.7}$$

Consequently, the joint near optimal importance is derived from the individual near optimal importance functions (7.5):

$$\begin{aligned}
 \hat{p}(X_t|X_{t-1}, y_t^d) &= \prod_{j=1}^{N_o} \hat{p}(x_t^j|x_{t-1}^j, y_t^d) \\
 &= \prod_{j=1}^{N_o} p(c_t^j|c_{t-1}^j, v_{t-1}^j, \tilde{s}_t^j, y_t^d) \cdot p(s_t^j|s_{t-1}^j) \cdot p(v_t^j|v_{t-1}^j)
 \end{aligned} \tag{7.8}$$

where $p(c_t^j|c_{t-1}^j, v_{t-1}^j, \tilde{s}_t^j, y_t^d)$ is given by expression (7.6).

Subsequently, the corresponding weight update is:

$$w_t \propto w_{t-1} \cdot \prod_{j=1}^{N_o} \frac{p(y_t|x_t^j) \cdot p(y_t^d|c_{t-1}^j, v_{t-1}^j, \tilde{s}_t^j)}{p(y_t^d|c_t^j, \tilde{s}_t^j)} \tag{7.9}$$

The resulting algorithm is summarised in Table 7.2.

<p>Initialisation ($t = 0$)</p> <p>sample $\{X_0^{(i)}\}_{i=1}^{N_p} \sim p(X_0)$</p> <p>initialise $\{w_0^{(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}$</p>
<p>Sequential processing ($t \geq 0$)</p> <p>for $i = 1 : N_p$</p> <div style="border-left: 1px solid black; padding-left: 10px;"> <p>for $j = 1 : N_o$</p> <div style="border-left: 1px solid black; padding-left: 10px;"> <p>sample $s_t^{j,(i)} \sim p(s_t^j s_{t-1}^{j,(i)})$</p> <p>sample $c_t^{j,(i)} \sim p(c_t^j c_{t-1}^{j,(i)}, v_{t-1}^{j,(i)}, \tilde{s}_t^j, y_t^d)$ using (7.6)</p> <p>sample $v_t^{j,(i)} \sim p(v_t^j v_{t-1}^{j,(i)})$</p> </div> <p>end for</p> <p>update $w_t^{(i)} = w_{t-1}^{(i)} \cdot \prod_{j=1}^{N_o} \frac{p(y_t c_t^{j,(i)}, s_t^{j,(i)}) \cdot p(y_t^d c_{t-1}^{j,(i)}, v_{t-1}^{j,(i)}, \tilde{s}_t^j)}{p(y_t^d c_t^{j,(i)}, \tilde{s}_t^j)}$</p> </div> <p>end for</p> <p>normalise the weights $w_t^{(i)}$</p> <p>estimate $\hat{X}_t = \sum_{i=1}^{N_p} w_t^{(i)} \cdot X_t^{(i)}$</p> <p>resample $\{X_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_p}$ to obtain $\{X_t^{(i)}, \frac{1}{N_p}\}_{i=1}^{N_p}$</p>

Table 7.2: Joint NOPF algorithm.

Finally, in the joint NOPF and the independent NOPFs, the sampling step is similar, the state space is explored by the individual near optimal importance functions. The difference is in the weighting step. In the joint configuration, a single weight measures the matching between the candidate regions proposed for all the objects and the observations. In the independent configuration, a weight is computed for the candidate region drawn for each object.

7.2.4 Experimental results

To evaluate the benefits of the near optimal approach in the context of MOT, we have conducted the following experimentations. We compare both tracking strategies: the joint PF and the independent PFs using two different importance functions: the Gaussian prior proposal and the near optimal proposal.

Two image sequences are considered. The BIWI sequence is composed of 81 images of size 720×576 and contains 7 non interacting targets. The ILAB sequence is composed of 171 images of size 704×576 and contains 5 interacting

7. Multiple Object Tracking



Figure 7.1: Image extracted from the BIWI sequence and the corresponding soft detection map.



Figure 7.2: Image extracted from the ILAB sequence and the corresponding soft detection map.

targets, these interactions cause multiple occlusions. These sequences are shown in Figures 7.1 and 7.2 respectively. The number of particles is $N_p = 100$ for each independent PF and $N_p = 2000$ for the joint PF.

7. Multiple Object Tracking

	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6	Target 7	Mean
Joint PF	86.79	64.07	85.31	81.11	66.79	83.58	89.63	79.61
Joint NOPF	76.79	66.17	87.90	88.40	71.48	87.41	94.32	81.78
Indep. PFs	63.74	88.84	87.02	97.83	84.59	95.63	98.72	88.05
Indep. NOPFs	69.28	90.81	95.84	99.42	90.86	97.64	99.44	91.90

Table 7.3: Success rate for the BIWI sequence.

Table 7.3 summarizes the success rate obtained by the four trackers on the BIWI sequence. These results show that on average the independent PFs per group perform better than the joint PF. This is due to the problem of high dimension of the joint PF. Also these results show that the use of the near optimal proposal allows to enhance the performance of both sampling strategies, since it offers a better state space exploration.

	Target 1	Target 2	Target 3	Target 4	Target 5	Mean
Joint PF	66.49	85.26	73.74	67.95	89.18	76.49
Joint NOPF	74.62	71.46	71.05	73.80	82.87	74.76
Indep. PFs	83.49	90.35	60.30	66.63	86.89	77.53
Indep. NOPFs	87.81	93.12	70.03	79.08	91.93	84.39

Table 7.4: Success rate for the ILAB sequence.

Table 7.4 summarizes the success rate obtained by the trackers on the ILAB sequence. Similarly to the BIWI sequence, the global performance of the independent PFs outperforms the joint PF. The near optimal proposal enhances the

performance of the independent PFs, but slightly degrades the performance of the joint PF. This degradation is due to the quality of the soft detection information. Indeed, in the ILAB sequence, people cross and overlap several times, and as no interaction is taken into account in the model, the overlapping is seen as noise. When two distinct objects overlap, the soft detection information coming from both objects form a single and undistinguishable mass, which may alter the particle drawing using the soft detection based near optimal proposal.

This problem gets worse in high dimension. In small dimension, if a part of the particle swarm is misguided by the presence of another moving object, the other part of the particle swarm is able to cover the real location of the object and then the state estimation is correct. But in high dimension, when drawing candidates for one object, if a part of the particles is driven into wrong areas of the state space, the number of effective particles intended to the other objects is significantly reduced. In other words, misguidance in high dimension is equivalent to sampling the same high dimensional state space with a smaller particle set. This phenomenon highlights a limitation of the near optimal proposal which becomes highly sensitive to detection noise (especially false positives) when the dimension of the state space grows.

Therefore it appears clearly that the main issue in multiple object tracking is the problem of high dimension. Although joint PFs allows a better modelling of complex scenarios with multiple interacting targets and thus offers possibilities that are beyond the reach of the "one PF for each target" solutions, the joint representation increases consequently the complexity of the problem, which threatens the efficiency and the robustness of the tracking.

The question of the high dimension in Monte Carlo methods has been studied in recent works [96] and scientific seminars¹. Among the proposed techniques, the local PF recently developed by Rebeschini et al. [84] seems to be a promising solution.

¹In the recent years, a seminar has been dedicated to this subject [44]

7.3 MOT using the local particle filter

To overcome the dimension problem in the joint configuration while taking into account the dependencies between the objects, we propose to use the local particle filter [84] which partitions the large state space into separate subspaces of smaller dimension.

7.3.1 Problem formulation - Tracking of multiple interacting objects

The multi-object model proposed in section 7.2.1 is adapted to take into account interactions between the objects. In visual tracking, the interactions are often modelled by a pairwise Markov random field motion prior [46, 119]. However the interaction term is too complex to be used in the sampling step, so it is only considered as an additional term in the importance weight or in the acceptance probability. Moreover, it reflects avoidance between the objects, which concerns top view 2D images as in [46] or 3D scenes reconstructed from multiple cameras as in [119]. In conventional 2D images, several objects can overlap because of the projection of the 3D scene on the 2D plane.

Here, we consider an other type of interaction. As in [56, 32, 5], the targets which are close to one another and move in the same direction tend to form a group and to adopt similar dynamics. The groups are supposed to evolve independently.

Then we consider that the N_o objects are divided into N_g independent groups $G_t = \{G_t^1, G_t^2, \dots, G_t^{N_g}\}$ with $\cup_{g=1}^{N_g} G_t^g = \{1 : N_o\}$. We denote $x_t(G_t^g)$ the set of the states of the objects belonging to the group G_t^g , thus $x_t(G_t^g) = \{x_t^j : j \in G_t^g\}$.

To model the interactions between the objects inside a group, the velocity of each object j is replaced by the average velocity of all the objects of the group G_{t-1}^g to which the object j belongs. Then the transition model becomes:

$$\begin{cases} c_t^j = c_{t-1}^j + v^{G_{t-1}^g} + \epsilon_c \\ v_t^j = v_{t-1}^j + \epsilon_v \\ s_t^j = s_{t-1}^j + \epsilon_s \end{cases} \quad (7.10)$$

where the group velocity $v^{G_{t-1}^g}$ is equal to:

$$v^{G_{t-1}^g} = \frac{1}{|G_{t-1}^g|} \cdot \sum_{k \in G_{t-1}^g} v_{t-1}^k$$

The state noises ϵ_c , ϵ_v , ϵ_s are independent white Gaussian noises with $\Sigma_c = \text{diag}(\sigma_c^2, \sigma_c^2)$, $\Sigma_v = \text{diag}(\sigma_v^2, \sigma_v^2)$ and $\Sigma_s = \text{diag}(\sigma_s^2, \sigma_s^2)$ the respective covariance matrices defining the uncertainty region around the previous states.

According to the state model, the prior density can be written as:

$$p(X_t|X_{t-1}) = \prod_{g=1}^{N_g} p(x_t(G_t^g)|X_{t-1}) \quad (7.11)$$

or as:

$$p(X_t|X_{t-1}) = \prod_{j=1}^{N_o} p(x_t^j|X_{t-1}) \quad (7.12)$$

As observations can come from the objects belonging to the same group, the likelihood is written as:

$$p(y_t|X_t) = \prod_{g=1}^{N_g} p(y_t|x_t(G_t^g)) \quad (7.13)$$

7.3.2 The local particle filter

The main idea of the local or block particle filter is to partition the state space into separate subspaces of small dimension, called blocks, and run one PF algorithm on each subspace.

The genesis of this method relies on the following observation: in high dimensional filtering models, a decay of correlation is generally observed between the regions of the state space which are distant enough from one another. Due to this decay of correlation property, the model is locally low dimensional, in the sense that the conditional distribution of the state only needs to be updated by observations located in a neighborhood. Rebeschini and al. propose to exploit this property to design local particle filters.

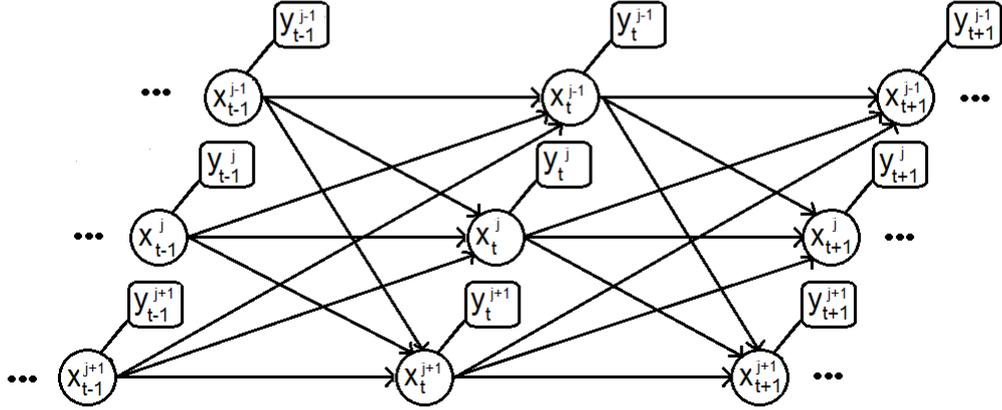


Figure 7.3: Dependence graph of a HMM satisfying the factorization (7.14).

Lets consider a HMM $(x_{1:t}, y_{1:t})_{t \geq 0}$ such as, at each time step t , the state x_t with dimension d can be divided into B_t independent and non overlapping subsets or blocks $\{D_t^g\}_{g=1}^{B_t}$. These blocks verify $\cup_{g=1}^{B_t} D_t^g = \{1 : d\}$ and $D_t^g \cap D_t^{g'} = \emptyset \forall g, g' \in \{1 : B_t\}$ with $g \neq g'$.

We assume that the HMM satisfies the following factorization:

$$p(y_t|x_t) \cdot p(x_t|x_{t-1}) = \prod_{g=1}^{B_t} f_t^g(y_t, x_{t-1}, x_t(D_t^g)) \quad (7.14)$$

where $x_t(D_t^g)$ is the set of the state components belonging to the subset D_t^g , thus $x_t(D_t^g) = \{x_t^j : j \in D_t^g\}$. The dependence graph of a HMM satisfying this factorization is illustrated in Figure 7.3. The graph shows that the dynamics of the state x_t is local:

$$p(x_t|x_{t-1}) = \prod_{g=1}^{B_t} p(x_t(D_t^g)|x_{t-1}) \quad (7.15)$$

Similarly, the observations are local. The likelihood, that is the conditional

distribution of y_t given all the states x_t , can be decomposed on the subsets and is written as:

$$p(y_t|x_t) = \prod_{g=1}^{B_t} p(y_t|x_t(D_t^g)) \quad (7.16)$$

To implement the local PF, a proposal density with a similar form is used:

$$q(x_t|x_{t-1}) = \prod_{g=1}^{B_t} q(x_t(D_t^g)|x_{t-1}) \quad (7.17)$$

The importance weights are also locally computed on each subset. The resulting algorithm is shown in Table 7.5.

Finally by running a PF on each non overlapping subset, the local PF approximates the filtering distribution as a product of marginal distributions on the B_t subsets:

$$p(x_t|y_{1:t}) \approx \bigotimes_{g=1}^{B_t} p(x_t(D_t^g)|y_{1:t}) \quad (7.18)$$

This strategy introduces some bias in the Monte Carlo estimation, as the approximation (7.18) does not converge to the exact filtering distribution as the number of particles tend to infinity. Nevertheless, the variance reduction due to the small dimension of the subsets is significant compared to the small amount of bias which is introduced [96].

<p>Initialisation (t=0) sample $\{x_0^{(i)}\}_{i=1}^{N_p} \sim p(x_0)$ initialise $\{w_0^{g,(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}, \forall g = 1, \dots, B_t$</p>
<p>Sequential processing (t>0) for $g = 1 : B_t$ for $i = 1 : N_p$ do sample $x_t^{(i)}(D_t^g) \sim q(x_t(D_t^g) x_{t-1})$ evaluate $w_t^{g,(i)} = \frac{p(y_t x_t^{(i)}(D_t^g))p(x_t^{(i)}(D_t^g) x_{t-1}^{(i)})}{q(x_t^{(i)}(D_t^g) x_{t-1}^{(i)})}$ end for normalise the importance weights $w_t^{g,(i)}$ end for $\hat{p}(x_t y_{1:t}) = \bigotimes_{g=1}^{B_t} \sum_{i=1}^{N_p} w_t^{g,(i)} \cdot \delta(x_t(D_t^g) - x_t^{(i)}(D_t^g))$ resample $\{x_t^{g,(i)}, w_t^{g,(i)}\}_{i=1}^{N_p} \forall g = 1, \dots, B_t$</p>

Table 7.5: Local particle filter algorithm.

Implementation for visual MOT

To apply the local particle filter to the visual tracking problem, we compare the MOT model described in section 7.3.1 with the model taken into account in the local PF.

If we assume that the likelihood is the product of the N_o individual object likelihoods: $p(y_t|X_t) = \prod_{j=1}^{N_o} p(y_t|x_t^j)$, then since the prior density ((7.12) is written as: $p(X_t|X_{t-1}) = \prod_{j=1}^{N_o} p(x_t^j|X_{t-1})$, we can implement the local PF by considering each object j as a subset or a block. That means the dimension of the estimation problem is divided by the number of objects.

The algorithm obtained by using the prior as the importance function is summarized in Table 9.5. In this work, we assume that the groups are known, but they could be evaluated jointly with the states of the objects, as in [32, 5].

<p>Initialisation (t=0) sample $\{X_0^{(i)}\}_{i=1}^{N_p} \sim p(X_0)$ initialise $\{w_0^{j,(i)}\}_{i=1}^{N_p} = \frac{1}{N_p}, \forall j = 1 : N_o$</p>
<p>Sequential processing (t>0) for $j = 1 : N_o$ for $i = 1 : N_p$ do sample $c_t^{j,(i)} \sim p(c_t^j c_{t-1}^{j,(i)}, v^{G_t^g})$ sample $s_t^{j,(i)} \sim p(s_t^j s_{t-1}^{j,(i)})$ sample $v_t^{j,(i)} \sim p(v_t^j v_{t-1}^{j,(i)})$ evaluate $w_t^{j,(i)} = p(y_t x_t^{j,(i)})$ end for normalise the importance weights $w_t^{j,(i)}$ end for estimate $\hat{X}_t = \bigotimes_{j=1}^{N_o} \sum_{i=1}^{N_p} w_t^{j,(i)} \cdot x_t^{j,(i)}$ resample $\{x_t^{j,(i)}, w_t^{j,(i)}\}_{i=1}^{N_p}, \forall j = 1 : N_o$ update the object groups $\{G_{t-1}^g\}_{g=1}^{N_g}$ update the group velocity $v^{G_t^g} = \frac{1}{ G_t^g } \cdot \sum k \in G_t^g v_t^k, \forall g = 1 : N_g$</p>

Table 7.6: Local particle filter algorithm for visual tracking.

7.3.3 Experimental results

To show the relevance of the local PF against the curse of dimensionality, several simulations have been conducted on four synthetic image sequences, each with 100 images and a size 500×500 . In the sequences, each object is represented by a rectangular patch with a specific color. The sequences S1, S2 and S3 contain 4 objects forming three separate groups of size (2,1,1) and the sequence S4 contains 6 objects forming three separate groups of size (4,1,1). The difference between these four sequences is the movement of the objects, ranging from deterministic dynamics (S1) to erratic dynamics (S3), while S4 offers a different group composition. The number of groups and their composition is known and does not change in time.

Three trackers are considered for this experiment :

7. Local Particle Filter

- The joint PF based tracker, which defines a joint state space model.

- The tracker based on multiple independent PFs per group, which defines a state space per group.

- The local PF, which considers each object as a subset of the joint state space model.

All the trackers take into account the interactions between the objects belonging to the same group. They use the prior density as the importance function. The number of particles are respectively 20 , 20 and 50 for the local, independent and joint PFs. The performance results are expressed as an average F-measure over the whole sequence and are averaged over 100 iterations.

	Joint PF	Independent PFs per group	Local PF
S1	78.54	90.89	93.10
S2	76.24	89.24	92.60
S3	77.68	90.38	92.94
S4	74.90	86.18	93.87

Table 7.7: Average F-measure for the synthetic sequences.

Table 7.7 shows the average F-measure obtained by the three trackers on the four synthetic sequences. The results show that the local PF achieves the best results in all the scenarios. Its superiority is due to its capability to handle interactions between objects while reducing the state space of dimension N_o to a one dimensional state space. The multiple independent PFs per object group have also good performance, but lower than the local PF. These performances are explained by switching from the state space of dimension N_o to a state space per group with smaller dimension. Finally the joint PF has the lowest performances as it applies no dimension reduction and directly deals with the high dimensional

state space.

If we look closer to the performance difference between the three trackers, we observe that for sequences S1, S2, S3, the performance obtained with the independent PFs per object group is slightly below the performance of the local PF. For the sequence S4, the F-measure difference between the independent PFs and the local PF becomes more important and the results of the independent PFs gets closer to the performance of the joint PF. This is due to the increase of the number of objects within one group in sequence S4. Four of the six objects belong to the same group while the maximum group size in S1, S2 and S3 is of two objects. The dimension of the state space corresponding to the group of four objects is significant and the independent PFs per group suffer from the same dimension problem as the joint PF. The major limitation of the tracker based on the independent PFs is that the performance depends on the size of the object groups. This limitation is overcome by the local PF.

These results confirm the previous observations on the high dimension limitation of particle filtering, which is one of the major issues when performing multiple object tracking. They also highlight the benefits of the local PF to overcome this limitation. The local PF takes into account all the interactions between the objects in a simple and robust framework which is affected neither by the number of objects in the scene nor the size of the object groups.

7.4 Chapter Conclusion

In this chapter, we extended the near optimal approach to multiple object tracking to improve the exploration of the state space. We proposed a near optimal solution for both the joint and independent sampling strategies. Experimental simulations on real scenarios highlight the relevance of the soft detection based near optimal proposal in MOT context, but also the sensitivity to the soft detection noise generated by the multiple objects.

Furthermore the simulations pointed out the impact of the high dimensionality on the performance of SMC methods. To address this problem, we proposed to

7. Local Particle Filter

use the local particle filter which allows to combine the interaction modelling and the partition of the large state space into separate subspaces of smaller dimension. Simulations on synthetic images show the efficiency of the local PF to improve the tracking performance in high dimensional problems.

Chapter 8

Conclusions and perspectives

The aim of the thesis is to enhance the state space exploration in visual tracking. For that purpose we derived near optimal proposals using the soft detection information.

In Chapter 2, we reviewed some of the main tracking methods and presented a classification of the different tracking schemes. In this work, we opted for a statistical formulation of the estimation problem in the Bayesian framework. This choice is motivated by the kernel based representation in one hand, and the capabilities of the sequential Monte Carlo (SMC) methods to adapt to various tracking schemes and to handle complex non linear models.

In Chapter 3, we presented the two main families of sequential Monte Carlo methods: particle filters (PFs) and Markov chain Monte Carlo (MCMC) methods.

PFs suffer from a weight degeneracy issue, related to the increase of the variance of importance weights over time. We showed how this problem is limited using resampling schemes, the auxiliary PF and the marginal PF. We also pointed out the importance of the importance function.

Concerning MCMC methods, we presented the Metropolis-Hastings algorithm and underlined the importance of the choice of the proposal that ensures a good state space exploration and a fast convergence of the Markov chain.

In Chapter 4, we explained how SMC algorithms can be applied to address

visual tracking problems, also we gave an overview of the state-of-the-art choices for the models and densities. In our implementation, we propose to enrich the observation model with a soft detection information, which is richer and more trustworthy than the usual binary detection output. Using this soft detection information is interesting to build proposals more efficient than the prior density.

The choice of the proposal is of crucial importance for the success of the Monte Carlo estimation. The purpose of the proposal is to guide the particles in the most likely areas of the state space. If the particles are propagated in inappropriate regions, the performance of the algorithms deteriorates significantly and, as stated previously, the impact in PFs is the weight degeneracy, while the convergence speed is impacted in the MCMC context.

The optimal proposal accounts for both the previous state and the current observation. Chapter 5 presented the main contribution of the thesis: the near optimal proposal. We proposed new proposal distributions based on the soft detection information and derived directly from the approximation of the optimal proposals using a simplifying hypothesis. This hypothesis allows to considerably reduce the computational effort necessary to compute the optimal proposals and thus to achieve a trade-off between optimality and computational cost. The near optimal proposals enhance the state space exploration with reasonable computation. We extended the near optimal approach to the auxiliary PF, marginal PF and sequential MCMC methods.

Enhancing the state space exploration in visual tracking is compulsory in two major situations: for abrupt motion tracking and multiple object tracking.

Chapter 6 showed the efficiency and the robustness of the near optimal methods in the context of abrupt motion on real image sequences representing different scenarios.

First we compared the near optimal PFs (NOPF) and near optimal sequential MCMC (NOMCMC) with the standard PFs and MCMC methods respectively. This experimentation showed the capabilities of the NOPF and NOMCMC to handle abrupt motion in real life scenarios even with a small number of particles.

Then, we compared between our near optimal PFs and sequential MCMC. The

experiment highlighted the interest of the particle pre selection of the NOAPF to further enhance the tracking performances in abrupt motion scenarios.

Finally we compared our methods with state-of-the-art methods developed specifically to handle abrupt motion. The simulations were conducted on three abrupt motion scenarios including low frame rate video, sudden dynamic changes and camera switching. They proved that our near optimal trackers are very competitive, robust and trustworthy against abrupt motion.

All these good results highlight the benefits of the soft detection information which improves the exploration of the state space by propagating samples in the most likely areas of the state space.

Multiple object tracking (MOT) also requires an effective proposal. As the problem dimension grows with the increase of the number of objects, there is a necessity to enhance the state space exploration.

In Chapter 7, the near optimal approach was extended to MOT in both joint and independent configuration.

Both near optimal multiple object trackers were tested on real image sequences. The simulations showed that the tracking performances are improved in MOT context in both configurations. The growth of the state space dimension also pointed out a limitation of the near optimal approach, which becomes more sensitive to noise in the soft detection information.

To go a step further, Chapter 7 presented the local PF which enables to take into account the interactions between the targets while partitioning the large state space into separate subspaces of smaller dimension. This filter was described, applied to MOT and tested on synthetic image sequences. The simulations showed the capability of the local PF to improve the tracking performance in high dimensional problems.

Perspectives

The proposed near optimal approach showed its relevance to enhance the state space exploration in visual tracking. These promising results encourage us to pursue this work and to consider the following axes of development.

A general optimisation of the source codes of our algorithms is needed to enhance the competitiveness of the proposed near optimal trackers in term of computational time.

Furthermore our last contributions, developed in the last chapter of this thesis and related to MOT, deserve to be further explored. Indeed the encouraging results that we have obtained require advanced experimentations.

For instance, as the MCMC methods are known to perform better than PFs in high dimensional problems, better results can be expected from the extension of the near optimal approach, in MOT context, to the MCMC framework.

In order to highlight the importance of interaction modelling in MOT, it is necessary to test the near optimal joint and independent PFs in tracking scenarios where the interactions between the objects are significant and have a strong impact on the tracking process.

Moreover the local PF should be tested on real image sequences to evaluate the benefits on MOT in real life scenarios. The extension of the near optimal approach to the local PF should result in a tracker capable of handling multiple interacting objects and robust against the difficulties of visual tracking scenarios, such as abrupt motion.

Publications

- Ameziane, M. O., Garnier, C., Delignon, Y., Duflos, E., Septier, F. Particle filtering with a soft detection based near-optimal importance function for visual tracking. In Signal Processing Conference (EUSIPCO), 2015 23rd European (pp.594-598). IEEE. Aout 2015, Nice, France.

-
- Mehdi Oulad Ameziane, Christelle Garnier, Yves Delignon, Emmanuel Duflos, François Septier. Filtrage particulaire avec une loi de proposition quasi-optimale utilisant la détection souple pour le suivi visuel. XXVème Colloque GRETSI, Sep 2015, Lyon, France. <hal-01198420>
 - Mehdi Oulad Ameziane, Christelle Garnier, François Septier, Emmanuel Duflos. Visual tracking of multiple objects using a local particle filter. GRETSI 2017 (Under review)

Chapter 9

Résumé en Français

9.1 Contexte de la thèse

Les travaux de cette thèse visent à améliorer l'exploration de l'espace d'état, dans le cadre des méthodes de Monte Carlo pour le suivi visuel.

Le suivi visuel est l'une des opérations les plus fondamentales dans le domaine de la vision par ordinateur. Le suivi est utile dans de nombreuses applications telles que la vidéo surveillance intelligente, le contrôle du trafic routier, les interactions homme-machine, les véhicules intelligents, etc.

L'objectif du suivi visuel est de reconstituer la trajectoire des objets d'intérêt dans une séquence d'images. Appliqué sur des images réelles, le suivi visuel s'avère une opération assez difficile car plusieurs facteurs (tels que les changements d'apparence et d'illumination, les occultations, les mouvements abrupts, etc.) perturbent le processus et provoquent des échecs. Afin de dépasser ces limitations, de nombreuses solutions ont été développées. Ces méthodes se divisent en deux grandes familles: les méthodes déterministes qui localisent l'objet par un processus d'optimisation tel que la descente de gradient; et les méthodes statistiques qui modélisent le problème de suivi dans le cadre Bayésien. Nous avons opté, dans cette thèse, pour la modélisation statistique, notamment le filtrage particulaire, qui est aujourd'hui largement utilisé et qui est plus adapté pour

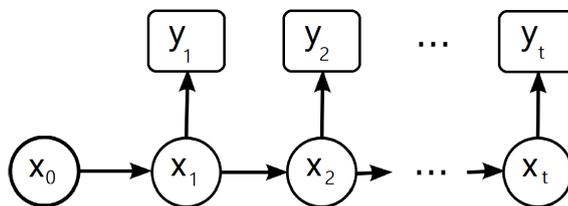


Figure 9.1: Graphe des dépendances dans un modèle de Markov caché.

gérer les incertitudes des modèles.

Dans ce contexte, nous modélisons le problème de suivi visuel suivant un modèle de Markov caché où les paramètres inconnus de l'objet x_t , incluant la position c_t et la taille de l'objet s_t , sont estimés de façon séquentielle à partir de la densité de filtrage $p(x_t|y_{1:t})$, avec $y_{1:t}$ l'ensemble des observations (voir figure 9.1). Cette densité de filtrage s'écrit suivant la loi de Bayes:

$$p(x_t|y_{1:t}) \propto p(y_t|x_t) \cdot \int p(x_t|x_{t-1}) \cdot p(x_{t-1}|y_{1:t-1}) dx_{t-1} \quad (9.1)$$

où $p(x_t|x_{t-1})$ est un loi a priori liée au modèle dynamique et $p(x_t|y_{1:t})$ la vraisemblance.

Dans le cas du suivi visuel on ne peut pas résoudre ce problème d'inférence de façon analytique, car le problème est non linéaire/non Gaussien, d'où le recours aux méthodes séquentielles de Monte Carlo (SMC). Ces méthodes permettent d'obtenir une approximation particulière de la densité de filtrage via un nombre N_p d'échantillons et leurs poids associés. Deux grandes familles de méthodes permettent d'obtenir l'approximation de Monte-Carlo: les filtres particuliers (PF) et les méthodes de Monte-Carlo par chaînes de Markov (MCMC).

Les méthodes de filtrage particulière connaissent une augmentation de la variance des poids d'importance au cours du temps. Ce phénomène, aussi appelé dégénérescence des poids, a un impact significatif sur les performances du PF. Diverses solutions ont été proposées pour palier ce problème, notamment l'introduction d'une routine de ré-échantillonnage, l'utilisation d'algorithmes alternatifs moins sensibles à la dégénérescence des poids tels que le PF auxiliaire et marginal. Enfin le choix de la loi de proposition est très important et a un

impact considérable sur la dégénérescence des poids. L'algorithme MCMC lui aussi connaît certaines limitations liées à l'exploration de l'espace d'état: si les déplacements proposés ne sont pas adaptés, la chaîne de Markov stagnera, ce qui nécessitera beaucoup d'itérations avant de converger vers la densité de filtrage.

De ce fait, le choix de la loi de proposition est un élément clé de la réussite de l'approximation particulière, et l'efficacité des méthodes de Monte Carlo en dépend fortement. Ainsi une loi de proposition adaptée à l'application permet d'explorer plus efficacement l'espace d'état, ce qui se traduit par une atténuation du phénomène de la dégénérescence des poids dans le cas du PF et une augmentation de la vitesse de convergence dans le cadre du MCMC. Le choix optimal pour la loi de proposition est la loi de filtrage, mais en principe celle-ci n'est pas disponible pour effectuer le tirage. Elle prends en compte les observations pour propager les particules.

9.2 État de l'art

Dans la littérature, de nombreuses stratégies sous optimales ont été proposées afin d'exploiter les observations courantes dans la loi de proposition.

Dans cadre du filtrage particulière, nous distinguons deux familles de méthodes :

- Les méthodes implicites qui propagent les particules suivant un modèle a priori puis réorientent les particules à travers une étape d'optimisation (mean shift, déplacement MCMC etc.).
- les méthodes explicites qui visent à construire une loi de proposition à partir des observations courantes par exemple sous forme d'un mélange entre la loi à priori et un mélange de Gaussiennes centrées sur des points spécifiques obtenus par détection.

Notre approche est plus directe, elle vise à obtenir une approximation de la loi de proposition optimale.

9.3 La loi de proposition "Near Optimal"

9.3.1 Loi de proposition optimale

Dans le cas du filtrage particulière, la loi de proposition optimale au sens de la minimisation de la variance des poids [22] prend en considération l'observation courante et s'écrit :

$$p(x_t|x_{t-1}, y_t) = \frac{p(y_t|x_t) \cdot p(x_t|x_{t-1})}{p(y_t|x_{t-1})} \quad (9.2)$$

Compte tenu de la non linéarité de la vraisemblance, cette loi de proposition ne peut pas être calculée de façon analytique et son évaluation point à point est extrêmement coûteuse en calculs. Le numérateur se compose de deux termes : la loi a priori qui est simple (marche aléatoire Gaussienne) et la vraisemblance qui est plus complexe. Dans la thèse nous cherchons à approcher la loi de proposition optimale via une approximation de la vraisemblance pour ainsi trouver un compromis entre optimalité et complexité.

9.3.2 Approximation de la loi de proposition optimale

Pour définir la vraisemblance, nous exploitons une information de détection souple y_t^d , obtenue par un algorithme de détection avant seuillage. Cette information, plus riche et plus fiable que le résultat de détection binaire, offre l'avantage d'une vraisemblance moins coûteuse en calculs que la vraisemblance couleur classique basée sur le calcul d'une distance entre histogrammes. De plus nous avons observé, pour la détection souple, que les variations de la taille des fenêtres avaient peu d'impact sur la localisation des modes de vraisemblance. Comme le montre la figure 9.2, la taille impacte plus l'étalement de la vraisemblance que la localisation du mode. Ainsi nous pouvons affirmer que pour localiser l'objet, en utilisant de détection souple, il n'est pas nécessaire de connaître exactement sa taille. Nous choisissons donc d'évaluer la vraisemblance pour une seule valeur de la taille :

$$\tilde{s}_t = E[s_t|\hat{s}_{t-1}]$$

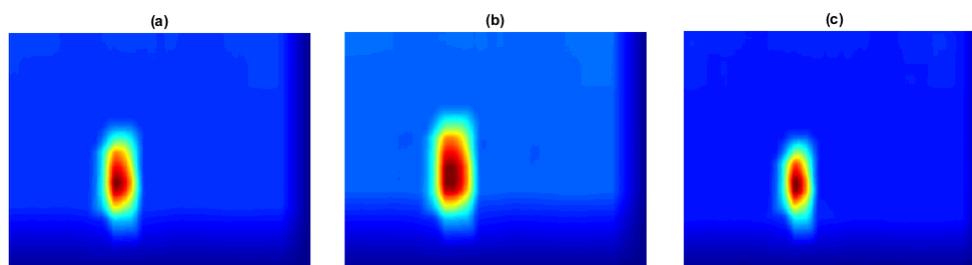


Figure 9.2: Vraisemblance de détection calculée pour différentes tailles de fenêtre: (a) taille réelle, (b) plus grande de 20% et (c) plus petite de 20%

<p>pour $i = 1 : N_p$</p> <p> tirer $s_t^{(i)} \sim p(s_t s_{t-1}^{(i)})$</p> <p> tirer $c_t^{(i)} \sim p(c_t c_{t-1}^{(i)}, \tilde{s}_t, y_t^d)$</p> <p> évaluer $w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{p(y_t c_t^{(i)}, s_t^{(i)}) \cdot p(y_t^d c_{t-1}^{(i)}, \tilde{s}_t)}{p(y_t^d c_t^{(i)}, \tilde{s}_t)}$</p> <p>fin pour</p> <p>normaliser les poids $w_t^{(i)}$</p> <p>ré-échantillonner</p>
--

Table 9.1: Near Optimal Particle Filter (NOPF)

Nous obtenons ainsi la loi de proposition dite "Near Optimal" :

$$\begin{aligned} \hat{p}(x_t | x_{t-1}, y_t^d) &= \frac{p(y_t^d | c_t, \tilde{s}_t) \cdot p(c_t | c_{t-1}) \cdot p(s_t | s_{t-1})}{p(y_t^d | c_{t-1}, \tilde{s}_t)} \\ &= p(c_t | c_{t-1}, \tilde{s}_t, y_t^d) \cdot p(s_t | s_{t-1}) \end{aligned}$$

Cette loi s'écrit comme le produit de deux lois: une approximation de la loi de proposition optimale pour le tirage de la position et la loi a priori pour la taille. Cette approximation permet de réduire significativement le temps de calcul nécessaire au tirage. L'algorithme de filtrage particulaire obtenu, appelé NOPF (Near Optimal Particle Filter) est résumé dans le tableau 9.1.

9.3.3 Variantes de l'algorithme NOPF

Nous avons étendu l'approche "Near Optimal" au filtre particulaire auxiliaire et aux méthodes MCMC séquentielles.

Plus précisément, nous avons étendu l'approche "near optimal" au filtre auxiliaire dit "Fully Adapted" qui, dans la terminologie de Pitt et al. [81] pré-sélectionne les particules à propager suivant la loi prédictive $p(y_t|x_{t-1})$ et effectue le tirage suivant la loi de proposition optimale $p(x_t|x_{t-1}, y_t)$. Pour la pré-sélection des particules, la loi prédictive est approché par l'expression suivante $p(y_t^d|c_{t-1}, \tilde{s}_t) = \sum_{c_t} p(y_t^d|c_t, \tilde{s}_t) \cdot p(c_t|c_{t-1})$ puis le tirage est effectué suivant la loi Near Optimal 9.3.1.

Nous avons aussi étendu notre approche à la méthode MCMC visant la densité de filtrage jointe $p(x_t, x_{t-1}|y_{1:t})$ développée par Septier et al. [95]. Le tirage est réalisé suivant la loi "Near Optimal" suivante:

$$\hat{p}(x_t, x_{t-1}|y_{1:t}) \propto p(y_t^d|c_t, \tilde{s}_t) \cdot p(x_t|x_{t-1}) \cdot \sum_{i=1}^{N_p} \delta(x_{t-1} - x_{t-1}^{(i)}) \quad (9.3)$$

A partir de ces lois de proposition "Near Optimal", nous avons implémenté les algorithmes correspondants, Notés NOAPF et NOMCMC.

L'améliorer l'exploration de l'espace d'état est indispensable dans deux applications courantes du suivi visuel:

- Le suivi des mouvements abrupts où les déplacements sont importants et imprévisibles entre deux images successives (lors de mouvements rapides, changements de dynamique, changements de camera, vidéos bas débit etc.). Dans ces cas le modèle a priori n'est pas suffisant pour assurer le suivi de l'objet.
- Le suivi multi-objets (MOT) qui se heurte à la problématique des espaces de grande dimension. Ceci se traduit par une croissance exponentielle du nombre de particules requis pour explorer l'espace d'état en fonction du nombre d'objets à suivre.

9.4 Application au suivi des mouvements abrupts

Afin de tester l'efficacité et la robustesse de nos algorithmes face aux mouvements abrupts, nous avons effectué plusieurs simulations. Principalement nous avons évalué les performances de l'approche "Near Optimal" face à des filtres particuliers plus standard puis face à des techniques de référence sur le suivi de mouvements abrupts. Ces simulations sont effectuées sur des séquences d'images réelles sous-échantillonnées avec un taux DS (Downsampling) pour accentuer les mouvements abrupts. Les performances sont indiquées sous la forme d'un taux de succès (Success rate), qui représente la moyenne des suivis réussis sur la séquence.

9.4.1 Comparaison du NOPF avec des PFs standards

Nous avons comparé 3 algorithmes: le NOPF, un PF standard [22] et un Boosted PF [72]. Ce dernier utilise une loi de proposition qui s'écrit sous la forme d'un mélange de Gaussiennes basé sur une information de détection dure.

Sur les 3 séquences de test notre algorithme obtient les meilleures performances. Les séquences 1 et 2 sont caractérisées par des mouvements rapides. Le PF standard a des difficultés à suivre l'objet quand la vitesse de déplacement est accentuée (DS rate =10).

Sur ces 3 séquences les informations de détection sont très bruitées, ce qui induit en erreur le Boosted PF et entraîne un suivi de qualité insuffisante. Seul le NOPF, utilisant une détection souple, plus fiable, non entachée d'erreurs de décision, parvient à suivre l'objet. Les résultats sont présentés dans les tableaux 9.2 et 9.3.

DS rate	PF	BPF	NOPF
5	86%	69%	100%
10	75%	42%	100%

Table 9.2: Success Rate obtenu pour la séquence 2.

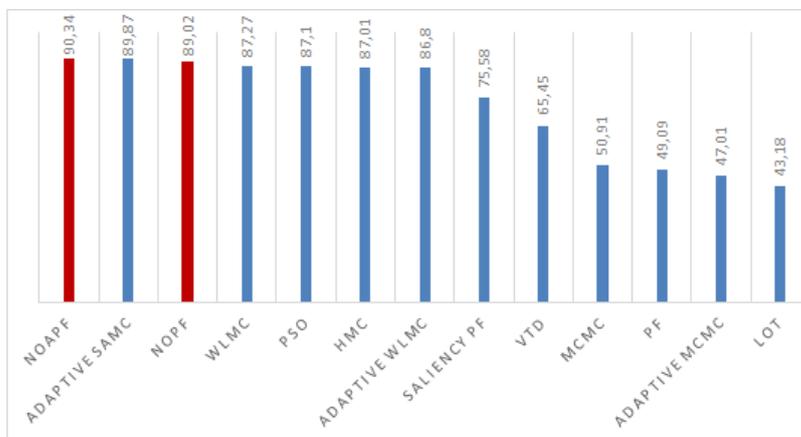


Figure 9.3: Success rate pour la séquence YoungKi.

DS rate	PF	BPF	NOPF
1	34%	41%	85%
10	27%	30%	71%

Table 9.3: Success Rate obtenu pour la séquence 3.

9.4.2 Comparaison avec des algorithmes de référence

Nous avons comparé nos algorithmes avec 11 autres algorithmes faisant partie de l'état de l'art en matière de suivi des mouvements abrupts. Ces comparaisons ont été réalisées sur des séquences d'images publiques, contenant des mouvements abrupts. Les résultats pour la séquence YoungKi sont présentés par la figure 9.3.

Nos algorithmes obtiennent les meilleurs résultats et notamment le filtre "Near Optimal Auxiliaire" qui, grâce à la présélection des particules, est le plus efficace de nos algorithmes "Near Optimal". En effet la présélection des particules, effectuée à la lumière des observations courantes, permet de mieux rattraper les discontinuités du mouvement abrupt. Les résultats sont similaires pour la séquence Ping-Pong Sur la séquence Tennis nous obtenons des bons résultats dans la moyenne de ceux obtenus par les concurrents, malgré la mauvaise qualité des images et de l'information de détection.

9.5 Application au suivi multi-objets

Le problème du suivi multi-objets (MOT) peut être résolu de deux manières: suivant une modélisation indépendante où un filtre est associé à chaque cible, ce choix permet de se ramener à un problème de petite dimension mais ne permet pas de prendre en considération les interactions entre cibles. La modélisation jointe permet de considérer un espace d'état joint incluant tous les objets, ce qui permet de modéliser les interactions entre objets. Mais elle est limitée par la malédiction de la grande dimension: le nombre de particules nécessaire pour réaliser une exploration satisfaisante de l'espace d'état croît exponentiellement avec la dimension.

Nous proposons premièrement d'étendre l'approche "Near Optimal" au suivi multi-objets suivant les deux modélisations précédentes. Puis nous proposons une approche alternative qui permet de conserver une configuration basse dimension tout en modélisant les interactions entre objets, en utilisant le Local PF [84].

9.5.1 Algorithme NOPF pour le suivi multi-objets

Pour le suivi d'objets multiples et indépendants, nous modélisons chaque objet j par $x_t^j = \{c_t^j, v_t^j, s_t^j\}$ où les variables sont respectivement la position, vitesse et taille de l'objet. Le vecteur d'état est $X_t = \{x_t^1, x_t^2, \dots, x_t^{N_o}\}$ avec N_o le nombre d'objets supposé fixe.

L'extension de la loi "Near Optimal" pour le suivi multi-objets, dans une modélisation indépendante, est obtenu directement en implémentant un filtre par objet. Par contre, dans une modélisation jointe, on approche la loi optimale jointe $p(X_t|X_{t-1}, y_t) = \prod_{j=1}^{N_o} p(x_t^j|x_{t-1}^j, y_t)$ en introduisant les mêmes hypothèses sur l'utilisation de la vraisemblance de détection souple pour une seule valeur de la taille. On obtient ainsi l'approximation suivante :

$$\hat{p}(X_t|X_{t-1}, y_t^d) = \prod_{j=1}^{N_o} p(c_t^j|c_{t-1}^j, v_{t-1}^j, \tilde{s}_t^j, y_t^d) \cdot p(s_t^j|s_{t-1}^j) \cdot p(v_t^j|v_{t-1}^j) \quad (9.4)$$

Nous avons testé ces algorithmes sur des séquences d'images réelles. Le tableau 9.4 montre que globalement la modélisation indépendante offre de meilleures performances et souligne l'apport de la loi de proposition "Near Optimal" dans le cas indépendant. Dans le cas joint, les performances du "Near Optimal" sont dégradées. Ceci met en évidence la problématique de grande dimension qui dégrade les performances globales du suivi et rend l'approche "Near Optimal" plus sensible.

	Success rate
PF Joint	76.49
NOPF Joint	74.76
PFs Indép.	77.53
NOPFs Indép.	84.39

Table 9.4: Success Rate obtenu pour le suivi multi-objets

Pour dépasser cette limitation, nous proposons d'implémenter le Local PF qui est spécifiquement conçu pour répondre à cette problématique de grande dimension.

9.5.2 Local PF pour le suivi multi-objets

Dans les espaces de grande dimension, on observe une baisse de corrélation entre des régions distantes de l'espace d'état. Le Local PF proposé par Rebeschini et al. [84] exploite cette propriété et subdivise l'espace d'état en blocs. Il permet ainsi de résoudre le problème dans un modèle qui est localement à faible dimension tout en modélisant les interactions entre blocs.

On suppose ainsi que l'espace d'état peut être divisé en B_t blocs disjoints et indépendants puis, en exécutant un filtre particulière par bloc, on approche la loi de filtrage par $p(x_t|y_{1:t}) \approx \bigotimes_{g=1}^{B_t} p(x_t(g)|y_{1:t})$. Cette approximation introduit un biais, mais le gain en réduction de la variance est significatif.

Pour tester l'efficacité du Local PF, nous l'avons implémenté en considérant que chaque objet constitue un bloc, en utilisant un modèle à vitesse de groupe. Les objets appartenant au même groupe adoptent une même dynamique traduite

par une vitesse de groupe. Ceci permet d'introduire des interactions entre objets. L'algorithme obtenu est résumé dans le tableau 9.5.

```

pour  $j = 1 : N_o$ 
  pour  $i = 1 : N_p$ 
    tirer  $c_t^{j,(i)} \sim p(c_t^j | c_{t-1}^{j,(i)}, v_{t-1}^{G^g,(i)})$ 
    tirer  $s_t^{j,(i)} \sim p(s_t^j | s_{t-1}^{j,(i)})$ 
    tirer  $v_t^{j,(i)} \sim p(v_t^j | v_{t-1}^{j,(i)})$ 
    évaluer  $w_t^{j,(i)} = p(y_t | x_t^{j,(i)})$ 
  fin pour
  normaliser les poids  $w_t^{j,(i)}$ 
fin pour
estimer  $\hat{X}_t = \bigotimes_{j=1}^{N_o} \sum_{i=1}^{N_p} w_t^{j,(i)} \cdot x_t^{j,(i)}$ 
ré-échantillonner  $\{x_t^{j,(i)}, w_t^{j,(i)}\}_{i=1}^{N_p}, \forall j = 1 : N_o$ 
màj des groupes  $\{G_t^g\}_{g=1}^{N_g}$ 
màj des vitesses de groupe  $v^{G_t^g,(i)} = \frac{1}{|G_t^g|} \cdot \sum_{k \in G_t^g} v_t^{k,(i)}, \forall g = 1 : N_g$ 

```

Table 9.5: Local PF pour le suivi multi-objets

Nous avons évalué les performances du Local PF sur des séquences d'images synthétiques représentant différents types de mouvements et de compositions de groupes. Les résultats sont présentés, dans le tableau 9.6, sous la forme d'un taux de recouvrement entre les fenêtres réelles et estimées.

Séquence	PF Joint	PFs Indép. par groupe	Local PF
S1	78.54	90.89	93.10
S2	76.24	89.24	92.60
S3	77.68	90.38	92.94
S4	74.90	86.18	93.87

Table 9.6: Taux de recouvrement moyen pour les séquences synthétiques

Les résultats obtenus soulignent l'intérêt du Local PF qui n'est pas impacté par la grande dimension de l'espace d'état et qui fournit de très bonnes performances tout en prenant en compte les interactions entre objets.

9.6 Conclusion

Dans ces travaux de thèse, nous avons proposé une nouvelle loi de proposition "Near Optimal" qui est dérivée directement de la loi de proposition optimale et qui représente un bon compromis entre efficacité et complexité de calcul. Nous avons aussi étendu cette approche au filtre particulaire auxiliaire et au MCMC séquentiel. Dans le premier cas, la présélection des particules rend notre approche encore plus performante. Nous avons par la suite testé nos algorithmes sur des scénarios réels et complexes, notamment les mouvements abrupts et le suivi multi-objets. Les résultats obtenus montrent clairement l'intérêt de notre approche qui est très compétitive, et mettent aussi en évidence la problématique des espaces de grande dimension. Nous avons enfin apporté une réponse à cette problématique de la dimension à travers l'usage du Local PF dont les résultats sont très encourageants.

Appendix A

Performance comparison between different implementations of the NOMCMC and NOMPF algorithms

In order to choose the most interesting implementation of our Near Optimal MCMC and Marginal PF algorithms we tested the two different versions presented in Chapter 5 on the Lemming 1 image sequence.

For the Marginal near optimal PFs we note V1 the first version of the NOMPF presented in section 5.3.3.1, this algorithm is summarised in Table (5.3). Also we note V2 the second version of the NOMPF presented in section 5.3.3.2, this algorithm is summarised in Table (5.4).

For the near optimal MCMC we note V1 the first version of the NOMCMC presented in section 5.4.1, this algorithm is summarised in Table (5.5). Also we note V2 the second version of the NOMPF presented in section 5.4.2, this algorithm is summarised in Table (5.6).

We look for average F-score, success rates, location error in pixels and running time of these algorithms. We obtain the following results.

Comparison Marginal NOPF

Algorithm \ Np	30	50	100	200
V1	77.15	82.50	84.89	85.89
V2	78.66	82.91	85.49	85.99

Table 7: F-score Vs Np for two versions of NOMPF

Algorithm \ Np	30	50	100	200
V1	8.74	8.12	6.85	6.34
V2	9.16	7.58	6.74	6.27

Table 8: Location error (in pixel) Vs Np for two versions of NOMPF

Algorithm \ Np	30	50	100	200
V1	81.13 %	95.75 %	97.5 %	99.25 %
V2	92 %	96.38 %	98.38 %	99.25 %

Table 9: Success Rate Vs Np for two versions of NOMPF

Algorithm \ Np	30	50	100	200
V1	-	1.65	3.40	9.36
V2	-	1.30	1.98	3.43

Table 10: Frame time Vs Np for two versions of NOMPF

The results above shows that the Near Optimal Marginal PF V2 is clearly better than the V1. Even if the gape between the two versions is getting smaller as the number of particles increases, the execution time of the V2 is clearly more interesting.

Comparison MCMC NOPF

Algorithm \ Np	30	50	100	200
V1	70.22	75.21	82.05	83.86
V2	68.88	74.48	81.47	83.73

Table 11: F-score Vs Np for two versions of NOPF MCMC

Algorithm \ Np	30	50	100	200
V1	10.40	9.35	8.31	7.41
V2	10.43	9.46	8.50	7.62

Table 12: Location error (in pixel) Vs Np for two versions of NOMCMC

Algorithm \ Np	30	50	100	200
V1	84.88 %	89.38 %	95 %	96.25 %
V2	83.75 %	88.88 %	95.25 %	96.63 %

Table 13: Success Rate Vs Np for two versions of NOMCMC

Algorithm \ Np	30	50	100	200
V1	-	2.58	7.77	20.20
V2	-	2.52	5.33	7.85

Table 14: Frame time Vs Np for two versions of NOMCMC

From this results we see that the two versions of the MCMC algorithm are quite similar in term of performances. The NOMCMC V1 has a slightly higher results in terms of F-score and location error but the computational cost of this version is much higher than the NOMCMC V2 especially when Np is high. Also if we look at the tracking Success rate, we see that the V2 is getting more and more interesting as the number of particles increases.

For this reasons we have choosen to use only the NOMCMC V2 and Near Optimal Marginal PF V2 for the rest of this thesis.

References

- [1] Christophe Andrieu, Éric Moulines, et al. On the ergodicity properties of some adaptive MCMC algorithms. *The Annals of Applied Probability*, 16(3):1462–1505, 2006. [40](#)
- [2] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002. [55](#)
- [3] D Avitzour. Stochastic simulation Bayesian approach to multitarget tracking. *Radar, sonar and navigation IEEE proceedings.*, 142(2):41–44, 1995. [117](#)
- [4] Andrew Backhouse, Zulfiqar Hasan Khan, and Irene Yu-Hua Gu. Robust object tracking using particle filters and multi-region mean shift. In *Advances in Multimedia Information Processing-PCM 2009*, pages 393–403. Springer, 2009. [48](#)
- [5] Loris Bazzani, Matteo Zanotto, Marco Cristani, and Vittorio Murino. Joint individual-group modeling for tracking. *IEEE transactions on pattern analysis and machine intelligence*, 37(4):746–759, 2015. [127](#), [131](#)
- [6] Thierry Bouwmans, Fida El Baf, and Bertrand Vachon. Background modeling using mixture of Gaussians for foreground detection—a survey. *Recent Patents on Computer Science*, 1(3):219–237, 2008. [49](#)

- [7] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1820–1833, 2011. [3](#), [44](#), [49](#), [64](#), [117](#)
- [8] Ted J Broida and Rama Chellappa. Estimation of object motion parameters from noisy images. *IEEE transactions on pattern analysis and machine intelligence*, (1):90–99, 1986. [13](#)
- [9] Yizheng Cai, Nando de Freitas, and James J Little. Robust visual tracking for multiple targets. In *European Conference on Computer Vision*, pages 107–118. Springer, 2006. [117](#)
- [10] Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in Hidden Markov Models*, chapter Sequential Monte Carlo Methods, pages 209–250. Springer Series in Statistics. Springer New York, 2005. [23](#), [25](#), [26](#)
- [11] François Caron, Manuel Davy, Emmanuel Duflos, and Philippe Vanheeghe. Particle filtering for multisensor data fusion with switching observation models: Application to land vehicle positioning. *IEEE transactions on Signal Processing*, 55(6):2703–2719, 2007. [44](#)
- [12] James Carpenter, Peter Clifford, and Paul Fearnhead. Improved particle filter for nonlinear problems. *Radar, Sonar and Navigation IEEE Proceedings*, 146(1):2–7, 1999. [26](#)
- [13] Yunqiang Chen, Yong Rui, and Thomas S Huang. JPDAF based HMM for real-time contour tracking. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001.*, volume 1, pages I–543. IEEE, 2001. [117](#)
- [14] Kiam Choo and David J Fleet. People tracking using hybrid Monte Carlo filtering. In *Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001.*, volume 2, pages 321–328. IEEE, 2001. [44](#)

- [15] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002. [12](#), [48](#), [58](#)
- [16] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003. [14](#)
- [17] Dung Nghi Truong Cong, Louahdi Khoudour, Catherine Achard, and Amaury Flancquart. Adaptive model for object detection in noisy and fast-varying environment. In *Image Analysis and Processing–ICIAP 2011*, pages 68–77. Springer, 2011. [51](#), [90](#)
- [18] Dung Nghi Truong Cong, François Septier, Christelle Garnier, Louahdi Khoudour, and Yves Delignon. Robust visual tracking via MCMC-based particle filtering. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012*, pages 1493–1496. IEEE, 2012. [44](#), [47](#), [48](#), [50](#), [61](#)
- [19] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, volume 1, pages 886–893. IEEE, 2005. [49](#)
- [20] J. Deutscher and I. Reid. Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, 61:185–205, February 2005. [4](#), [58](#)
- [21] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69. IEEE, 2005. [25](#)
- [22] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000. [3](#), [28](#), [55](#), [56](#), [85](#), [144](#), [147](#)

REFERENCES

- [23] Arnaud Doucet, B-N Vo, Christophe Andrieu, and Manuel Davy. Particle filtering for multi-target tracking and sensor management. In *Proceedings of the Fifth International Conference on Information Fusion, 2002.*, volume 1, pages 474–481. IEEE, 2002. [117](#)
- [24] Séverine Dubuisson. Tree-structured image difference for fast histogram and distance between histograms computation. *Pattern Recognition Letters*, 32(3):411–422, 2011. [63](#)
- [25] Séverine Dubuisson and Christophe Gonzales. Min-space integral histogram. In *Computer Vision–ECCV 2012*, pages 188–201. Springer, 2012. [63](#)
- [26] Erkut Erdem, Séverine Dubuisson, and Isabelle Bloch. Visual tracking by fusing multiple cues with context-sensitive reliabilities. *Pattern Recognition*, 45(5):1948–1959, 2012. [51](#)
- [27] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE journal of Oceanic Engineering*, 8(3):173–184, 1983. [117](#)
- [28] Dieter Fox, Sebastian Thrun, Wolfram Burgard, and Frank Dellaert. Particle filters for mobile robot localization. In *Sequential Monte Carlo methods in practice*, pages 401–428. Springer, 2001. [59](#)
- [29] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995. [49](#)
- [30] Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984. [34](#)
- [31] Walter R Gilks and Carlo Berzuini. Following a moving target — Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146, 2001. [3](#), [57](#)

-
- [32] Amadou Gning, Lyudmila Mihaylova, Simon Maskell, Sze Kim Pang, and Simon Godsill. Group object structure and state estimation with evolving networks and Monte Carlo methods. *IEEE Transactions on Signal Processing*, 59(4):1383–1396, 2011. [127](#), [131](#)
- [33] Simon Godsill and Tim Clapp. Improvement strategies for Monte Carlo particle filters. In *Sequential Monte Carlo methods in practice*, pages 139–158. Springer, 2001. [3](#), [58](#)
- [34] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. 140(2):107–113, 1993. [53](#)
- [35] Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive metropolis algorithm. *Bernoulli*, pages 223–242, 2001. [40](#)
- [36] X Hai-Xia, W Yao-Nan, Z Wei, Z Jiang, and Y Xiao-Fang. Multi-object visual tracking based on reversible jump Markov chain Monte Carlo. *IET computer vision*, 5(5):282–290, 2011. [61](#), [118](#), [120](#)
- [37] W Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. [34](#)
- [38] Jeroen D Hol, Thomas B Schon, and Fredrik Gustafsson. On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, pages 79–82. IEEE, 2006. [25](#), [26](#)
- [39] Yufei Huang and Petar M Djuric. A hybrid importance function for particle filtering. *IEEE Signal Processing Letters*, 11(3):404–406, 2004. [56](#)
- [40] Carine Hue, J-P Le Cadre, and Patrick Perez. Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Transactions on signal processing*, 50(2):309–325, 2002. [117](#)
- [41] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *European conference on computer vision*, pages 343–356. Springer, 1996. [ix](#), [11](#)

-
- [42] Michael Isard and Andrew Blake. Condensation—conditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28, 1998. [12](#), [53](#)
- [43] Michael Isard and John MacCormick. Bramble: A Bayesian multiple-blob tracker. In *Eighth IEEE International Conference on Computer Vision, 2001. ICCV 2001.*, volume 2, pages 34–41. IEEE, 2001. [117](#)
- [44] GdR ISIS. Filtrage Bayésien en grande dimension par méthodes de Monte Carlo. [126](#)
- [45] V. John, Y. Trucco, A.S. Montemayor, and S. Ivekovic. Markerless human articulated tracking using hierarchical particle swarm optimisation. *Image and Vision Computing*, 28:1530–1547, November 2010. [4](#), [58](#)
- [46] Zia Khan, Tucker Balch, and Frank Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *IEEE transactions on pattern analysis and machine intelligence*, 27(11):1805–1819, 2005. [36](#), [44](#), [59](#), [86](#), [117](#), [118](#), [120](#), [127](#)
- [47] Mike Klaas, Nando De Freitas, and Arnaud Doucet. Toward practical n2 Monte Carlo: the marginal particle filter. *arXiv preprint arXiv:1207.1396*, 2012. [32](#)
- [48] Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288, 1994. [23](#)
- [49] Chris Kreucher, A Hero, and Keith Kastella. Multiple model particle filtering for multitarget tracking. In *the twelfth annual workshop on adaptive sensor array processing*. Lexington, MA, 2004. [117](#)
- [50] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. [13](#)
- [51] Junghyun Kwon, Kyoung Mu Lee, and Frank Chongwoo Park. Visual tracking via geometric particle filtering on the affine group with optimal impor-

-
- tance functions. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.*, pages 991–998. IEEE, 2009. [44](#)
- [52] Junseok Kwon and Kyoung Mu Lee. Tracking of abrupt motion using Wang-Landau Monte Carlo estimation. In *Computer Vision–ECCV 2008*, pages 387–400. Springer, 2008. [44](#), [46](#), [87](#)
- [53] Junseok Kwon and Kyoung Mu Lee. Visual tracking decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, pages 1269–1276. IEEE, 2010. [88](#)
- [54] Junseok Kwon and Kyoung Mu Lee. Wang-Landau Monte Carlo-based tracking methods for abrupt motions. *Transactions on Pattern Analysis and Machine Intelligence, IEEE*, 35(4):1011–1024, 2013. [46](#), [87](#), [107](#)
- [55] Oswald Lanz. Approximate bayesian multibody tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1436–1449, 2006. [117](#)
- [56] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011*, pages 120–127. IEEE, 2011. [127](#)
- [57] Yuan Li, Haizhou Ai, Takayoshi Yamashita, Shihong Lao, and Masato Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1728–1740, 2008. [44](#)
- [58] Faming Liang, Chuanhai Liu, and Raymond J Carroll. Stochastic approximation in Monte Carlo computation. *Journal of the American Statistical Association*, 102(477):305–320, 2007. [88](#)
- [59] Mei Kuan Lim, Chee Seng Chan, Dorothy Monekosso, and Paolo Remagnino. Refined particle swarm intelligence method for abrupt motion tracking. *Information Sciences*, 283:267–287, 2014. [89](#), [107](#)

-
- [60] Jun S Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6(2):113–119, 1996. [23](#)
- [61] Jun S Liu and Rong Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998. [25](#)
- [62] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. [48](#)
- [63] W. Lu, K. Okuma, and J.J. Little. Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image and Vision Computing*, 27:189–205, January 2009. [3](#), [4](#), [50](#), [59](#)
- [64] John MacCormick and Michael Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *European Conference on Computer Vision*, pages 3–19. Springer, 2000. [44](#), [118](#)
- [65] E. Maggio and A. Cavallaro. Hybrid particle filter and mean shift tracker with adaptive transition model. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 221–224. IEEE, 2009. [4](#), [58](#)
- [66] John Makhoul, Francis Kubala, Richard Schwartz, Ralph Weischedel, et al. Performance measures for information extraction. In *Proceedings of DARPA broadcast news workshop*, pages 249–252, 1999. [84](#)
- [67] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953. [34](#), [39](#)
- [68] Lyudmila Mihaylova, Avishy Y Carmi, François Septier, Amadou Gning, Sze Kim Pang, and Simon Godsill. Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking. *Digital Signal Processing*, 25:1–16, 2014. [60](#)

-
- [69] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. Multiple object tracking benchmark. [104](#)
- [70] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler. MOT16: A benchmark for multi-object tracking. *arXiv:1603.00831 [cs]*, March 2016. arXiv: 1603.00831. [104](#)
- [71] Katja Nummiaro, Esther Koller-Meier, and Luc Van Gool. Object tracking with an adaptive color-based particle filter. In *Joint Pattern Recognition Symposium*, pages 353–360. Springer, 2002. [47](#)
- [72] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004. [3](#), [4](#), [50](#), [59](#), [86](#), [147](#)
- [73] Shaul Oron, Aharon Bar-Hillel, Dan Levi, and Shai Avidan. Locally orderless tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012*, pages 1940–1947. IEEE, 2012. [88](#)
- [74] Mehdi Oulad Ameziane, Christelle Garnier, Yves Delignon, Emmanuel Duflos, and François Septier. Particle filtering with a soft detection based near-optimal importance function for visual tracking. In *Signal Processing Conference (EUSIPCO), 2015 23rd European*, pages 594–598. IEEE, 2015. [44](#), [49](#)
- [75] Sze Kim Pang, Jack Li, and Simon J Godsill. Models and algorithms for detection and tracking of coordinated groups. In *Aerospace Conference, 2008 IEEE*, pages 1–17. IEEE, 2008. [37](#)
- [76] Juan José Pantrigo, A Sánchez, Antonio S Montemayor, and Abraham Duarte. Multi-dimensional visual tracking using scatter search particle filter. *Pattern Recognition Letters*, 29(8):1160–1174, 2008. [58](#)
- [77] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Sixth international conference on Computer vision, 1998.*, pages 555–562. IEEE, 1998. [49](#)

-
- [78] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *European Conference on Computer Vision (ECCV)*, pages 661–675. 2002. [47](#)
- [79] Patrick Pérez and Jaco Vermaak. Bayesian tracking with auxiliary discrete processes. application to detection and tracking of objects with occlusions. In *Dynamical Vision*, pages 190–202. Springer, 2007. [44](#)
- [80] Patrick Perez, Jaco Vermaak, and Andrew Blake. Data fusion for visual tracking with particles. *Proceedings of the IEEE*, 92(3):495–513, 2004. [3](#), [4](#), [50](#), [51](#), [59](#)
- [81] Michael K Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American statistical association*, 94(446):590–599, 1999. [3](#), [29](#), [57](#), [86](#), [146](#)
- [82] Fatih Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, volume 1, pages 829–836. IEEE, 2005. [63](#)
- [83] Christopher Rasmussen and Gregory D. Hager. Probabilistic data association methods for tracking complex visual objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):560–576, 2001. [117](#)
- [84] Patrick Rebeschini, Ramon Van Handel, et al. Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability*, 25(5):2809–2866, 2015. [5](#), [7](#), [118](#), [126](#), [127](#), [149](#), [150](#)
- [85] Donald Reid. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 24(6):843–854, 1979. [117](#)
- [86] Gareth O Roberts, Andrew Gelman, Walter R Gilks, et al. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997. [40](#)

-
- [87] Gareth O Roberts and Jeffrey S Rosenthal. Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009. [4](#), [41](#), [60](#)
- [88] Gareth O Roberts and Richard L Tweedie. Bounds on regeneration times and convergence rates for Markov chains. *Stochastic Processes and their applications*, 80(2):211–229, 1999. [39](#)
- [89] Jeffrey S Rosenthal. Minorization conditions and convergence rates for Markov chain Monte Carlo. *Journal of the American Statistical Association*, 90(430):558–566, 1995. [39](#)
- [90] Jeffrey S Rosenthal. Quantitative convergence rates of Markov chains: A simple account. *Electronic Communications in Probability*, 7(13):123–128, 2002. [39](#)
- [91] Jeffrey S Rosenthal et al. Optimal proposal distributions and adaptive MCMC. *Handbook of Markov Chain Monte Carlo*, pages 93–112, 2011. [4](#), [39](#), [60](#), [87](#)
- [92] Sylvain Rousseau, Pierre Chainais, and Christelle Garnier. Dictionary learning for a sparse appearance model in visual tracking. In *IEEE International Conference on Image Processing (ICIP), 2015*, pages 4506–4510. IEEE, 2015. [47](#)
- [93] Yong Rui and Yunqiang Chen. Better proposal distributions: Object tracking using unscented particle filter. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–786. IEEE, 2001. [44](#)
- [94] F Septier, A Carmi, SK Pang, and SJ Godsill. Multiple object tracking using evolutionary and hybrid MCMC-based particle algorithms. In *15th IFAC Symposium on System Identification, 2009*, volume 15. IFAC, 2009. [61](#)
- [95] François Septier, Sze Kim Pang, Avishy Carmi, and Simon Godsill. On MCMC-based particle methods for Bayesian filtering: Application to mul-

-
- titarget tracking. In *3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009*, pages 360–363. IEEE, 2009. [37](#), [38](#), [60](#), [86](#), [118](#), [120](#), [146](#)
- [96] François Septier and Gareth W. Peters. An Overview of Recent Advances in Monte-Carlo Methods for Bayesian Filtering in High-Dimensional Spaces. In Gareth W. Peters and Tomoko Matsui, editors, *Theoretical Aspects of Spatial-Temporal Modeling*. SpringerBriefs - JSS Research Series in Statistics, Isbn 9784431553359, November 2015. [118](#), [126](#), [130](#)
- [97] François Septier and Gareth W Peters. Langevin and hamiltonian based sequential mcmc for efficient bayesian filtering in high-dimensional spaces. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):312–327, 2016. [61](#), [62](#)
- [98] Ishwar K Sethi and Ramesh Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Transactions on pattern analysis and machine intelligence*, (1):56–73, 1987. [13](#)
- [99] Khurram Shafique and Mubarak Shah. A noniterative greedy algorithm for multiframe point correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 27(1):51–65, 2005. [ix](#), [13](#)
- [100] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, and Mubarak Shah. Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1442–1468, 2014. [10](#)
- [101] Kevin Smith, Daniel Gatica-Perez, and J-M Odobez. Using particles to track varying numbers of interacting people. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, volume 1, pages 962–969. IEEE, 2005. [44](#)
- [102] Kevin C Smith. Order matters: a distributed sampling method for multi-object tracking. Technical report, IDIAP, 2004. [118](#)

- [103] Kevin Charles Smith. *Bayesian methods for visual multi-object tracking with applications to human activity recognition*. PhD thesis, Citeseer, 2007. [117](#)
- [104] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.*, volume 2, pages 246–252. IEEE, 1999. [49](#)
- [105] Yingya Su, Qingjie Zhao, Liujun Zhao, and Dongbing Gu. Abrupt motion tracking using a visual saliency embedded particle filter. *Pattern Recognition*, 47(5):1826–1834, 2014. [3](#), [4](#), [85](#), [88](#), [107](#)
- [106] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*, pages 2553–2561, 2013. [49](#)
- [107] Hisashi Tanizaki and Roberto S Mariano. Nonlinear filters based on Taylor series expansions. *Communications in Statistics-Theory and Methods*, 25(6):1261–1282, 1996. [58](#)
- [108] Péter Torma and Csaba Szepesvári. Enhancing particle filters using local likelihood sampling. In *European Conference on Computer Vision*, pages 16–27. Springer, 2004. [59](#)
- [109] Rudolph Van Der Merwe, Arnaud Doucet, Nando De Freitas, and Eric Wan. The unscented particle filter. In *Nips*, volume 2000, pages 584–590. Denver, CO, USA, 2000. [58](#)
- [110] Namrata Vaswani. Particle filtering for large-dimensional state spaces with multimodal observation likelihoods. *IEEE Transactions on Signal Processing*, 56(10):4583–4597, 2008. [117](#)
- [111] Jaco Vermaak, Simon J Godsill, and Patrick Perez. Monte Carlo filtering for multi target tracking and data association. *IEEE Transactions on Aerospace and Electronic Systems*, 41(1):309–332, 2005. [117](#)

-
- [112] Paul Viola, Michael J Jones, and Daniel Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005. [49](#)
- [113] Fasheng Wang, Xucheng Li, Mingyu Lu, and Zhibo Xiao. Robust abrupt motion tracking via adaptive hamiltonian monte carlo sampling. In *PRICAI 2014: Trends in Artificial Intelligence*, pages 52–63. Springer, 2014. [107](#)
- [114] Fasheng Wang and Mingyu Lu. Hamiltonian Monte Carlo estimator for abrupt motion tracking. In *21st International Conference on Pattern Recognition (ICPR), 2012*, pages 3066–3069. IEEE, 2012. [61](#), [62](#), [88](#)
- [115] Nicolas Widynski. *Intégration d’informations spatiales floues dans un filtre particulière pour le suivi mono-et multi-objets dans des séquences d’images 2D*. PhD thesis, Paris 6, 2010. [118](#)
- [116] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Visual tracker benchmark. [91](#)
- [117] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015. [10](#), [104](#)
- [118] Hanxuan Yang, Ling Shao, Feng Zheng, Liang Wang, and Zhan Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011. [10](#)
- [119] Jian Yao and Jean-Marc Odobez. Multi-person Bayesian tracking with multiple cameras. *Multi-Camera Networks, editeurs, H. Aghajan and A. Cavallaro,*, pages 363–388, 2009. [44](#), [51](#), [61](#), [118](#), [120](#), [127](#)
- [120] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13, 2006. [ix](#), [10](#), [14](#), [48](#)
- [121] Alper Yilmaz, Xin Li, and Mubarak Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on pattern analysis and machine intelligence*, 26(11):1531–1536, 2004. [12](#)

- [122] Tao Zhao and Ram Nevatia. Tracking multiple humans in crowded environment. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'04, pages 406–413, 2004. [61](#)
- [123] Tao Zhao, Ram Nevatia, and Bo Wu. Segmentation and tracking of multiple humans in crowded environments. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 30(7):1198–1211, July 2008. [61](#), [118](#), [120](#)
- [124] Tianfei Zhou, Yao Lu, Feng Lv, Huijun Di, Qingjie Zhao, and Jian Zhang. Abrupt motion tracking via nearest neighbor field driven stochastic sampling. *Neurocomputing*, 165:350–360, 2015. [104](#), [107](#)
- [125] Xiuzhuang Zhou and Yao Lu. Abrupt motion tracking via adaptive stochastic approximation Monte Carlo sampling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, pages 1847–1854. IEEE, 2010. [46](#), [88](#)
- [126] Xiuzhuang Zhou, Yao Lu, Jiwen Lu, and Jie Zhou. Abrupt motion tracking via intensively adaptive Markov-chain Monte Carlo sampling. *IEEE Transactions on Image Processing*, 21(2):789–801, 2012. [88](#), [107](#)
- [127] I. Zuriarrain, A.A. Mekonnen, F. Lerasle, and A. Nestor. Tracking-by-detection of multiple persons by a resample-move particle filter. *Machine Vision and Applications*, 24:1751–1765, November 2013. [3](#), [4](#), [44](#), [50](#), [57](#), [59](#)

Titre : Amélioration de l'exploration de l'espace d'état dans les méthodes de Monte-Carlo séquentielles pour le suivi visuel

Le suivi visuel est une tâche essentielle en vision par ordinateur. Les méthodes de Monte Carlo séquentielles (SMC) sont largement utilisées aujourd'hui pour résoudre la problématique du suivi visuel. L'efficacité des méthodes SMC dépend fortement du choix de la loi de proposition utilisée pour explorer l'espace d'état. Dans cette thèse, nous cherchons à améliorer l'exploration de l'espace d'état en approchant la loi de proposition optimale. La loi de proposition quasi-optimale proposée repose sur une approximation de la fonction de vraisemblance définie à partir d'une information de détection souple qui est à la fois plus fiable et moins coûteuse à calculer. En comparaison avec les travaux existants sur le sujet, cette loi de proposition quasi-optimale offre un bon compromis entre optimalité et complexité algorithmique.

L'amélioration de l'exploration de l'espace d'état est nécessaire dans deux applications du suivi visuel : le suivi des mouvements abrupts et le suivi multi-objets. Dans cette thèse nous avons montré la capacité des méthodes SMC quasi-optimales à suivre les mouvements abrupts, en les comparant aux méthodes spécifiquement conçues pour ce type de scénario. Nous avons aussi étendu la loi de proposition quasi-optimale au cas multi-objets. Enfin, nous avons implémenté le filtre particulaire local qui permet de partitionner l'espace d'état de grande dimension en sous-espaces de taille inférieure, tout en modélisant les interactions entre objets. Les résultats de simulation montrent l'intérêt d'exploiter l'information de détection souple dans la loi de proposition et prouvent que les algorithmes proposés améliorent la précision et la robustesse du suivi.

Mots clés : Monte Carlo Séquentiel, filtrage particulaire, MCMC, suivi visuel, détection souple, loi de proposition optimale, suivi multi-objets, filtre particulaire Local.

Title: Enhancement of the state space exploration in sequential Monte Carlo methods for visual tracking

Visual tracking is a fundamental task in computer vision applications. Sequential Monte Carlo (SMC) methods are widely used today to solve the tracking problem. The efficiency of SMC methods strongly depends on the choice of the proposal function which permits to explore the state space. In this thesis we aim to enhance the state space exploration by approximating the optimal proposal. The resulting near optimal proposal relies on an approximation of the likelihood function based on a soft detection information, which is more reliable and easier to compute. Our near optimal proposal offers a great trade-off between optimality and complexity.

Enhancing the state space exploration is necessary in two tracking scenarios: Abrupt motion and multi-object tracking. In this thesis we showed the capabilities of the near optimal SMC methods to handle abrupt movements in comparison with state of the art methods. We have also extended the near optimal proposal to the multi-object tracking framework. Finally we implemented the Local particle filter which permits the partitioning of the state space into smaller subspaces while handling interactions between objects. The results obtained highlights the interest of using the soft detection information in the proposal and shows the efficiency and the robustness of the proposed tracking algorithms.

Keywords: Sequential Monte Carlo, particle filtering, MCMC, visual tracking, soft detection, optimal proposal, multiple object tracking, local particle filter.