UNIVERSITÉ PIERRE ET MARIE CURIE - PARIS 6 ÉCOLE DOCTORALE EDITE

THÈSE

pour obtenir le titre de

Docteur en Sciences

de l'Université Pierre et Marie Curie - Paris 6 **Mention : INFORMATIQUE**

Présentée et soutenue par Nicolas WIDYNSKI

INTÉGRATION D'INFORMATIONS SPATIALES FLOUES DANS UN FILTRE PARTICULAIRE POUR LE SUIVI MONO- ET MULTI-OBJETS DANS DES SÉQUENCES D'IMAGES 2D.

Thèse dirigée par Séverine DUBUISSON et Isabelle BLOCH préparée au LTCI (Télécom ParisTech) soutenance prévue le 26 novembre 2010 devant le jury composé de

Jury :

<i>Rapporteurs</i> :	Mylène MASSON	-	UTC
	Wojciech PIECZYNSKI	-	Télécom SudParis
Examinateurs :	Elise ARNAUD	-	Université Joseph Fourier
	Maurice MILGRAM	-	ISIR
Invité :	Jacques BLANC-TALON	-	DGA
Directrices :	Séverine DUBUISSON	-	LIP6
	Isabelle BLOCH	-	Télécom ParisTech

Résumé

Dans cette thèse, nous nous intéressons au suivi d'objets par filtrage particulaire dans des séquences d'images 2D. En effet, les méthodes séquentielles de Monte-Carlo, ou filtres particulaires, ont connu un succès grandissant depuis la fin des années 1990 dans la communauté du suivi d'objets, de par leur faculté d'adaptation à ce type de problématique, leur efficacité en termes de performances ainsi que leur mise en œuvre souvent directe.

Dans ce contexte, nous étudions l'intégration d'informations floues dans le filtre particulaire, afin de guider le suivi d'objets en ajoutant des informations spatiales, qui peuvent être de type structurel, de forme, ou encore peuvent correspondre à des contraintes spatiales dans le cas multi-objets. Les informations spatiales floues se sont révélées très efficaces dans de nombreux domaines en traitement d'images (segmentation, reconnaissance...), mais n'ont, à notre connaissance, jamais été utilisées dans le cadre méthodologique du filtrage particulaire.

Nous établissons tout d'abord un modèle général pour la manipulation d'informations spatiales floues. Chaque relation spatiale ou caractéristique de forme est considérée comme une variable linguistique, prenant un petit nombre de valeurs. La sémantique de chacune des valeurs est donnée par un ensemble flou sur le domaine de la variable. Ainsi, par exemple, l'orientation d'un objet peut être décomposée selon un nombre fini d'ensembles flous : « nord », « nord-est », « est »... De manière similaire, le concept flou « forme de bouche » peut prendre des valeurs telles que « ouverte », « fermée », etc.

Les informations spatiales floues ainsi définies sont intégrées au filtre particulaire dans la dynamique de l'état inconnu, décrivant l'objet à suivre, du système. Cette approche permet d'ajouter une couche sémantique à la modélisation du problème de suivi. En intégrant dans l'espace d'état une information spatiale floue, cela permet de mieux guider les particules pendant la phase prédictive. Nous exploitons ensuite le fait que cette donnée n'évolue généralement que graduellement au cours du temps : nous favorisons, par exemple, l'orientation « nord-est » à perdurer dans le temps, tout en acceptant un changement vers un état voisin, comme le « nord » ou l' « est ». Ce type de modélisation permet, à la différence des méthodes classiques, d'intégrer au filtre particulaire une information synthétisée des k dernières estimations. Cela a l'avantage d'augmenter la robustesse du suivi face au bruit. De plus, pour prédire l'ensemble flou adéquat, nous utilisons également la vraisemblance afin de mieux gérer les changements brusques de la trajectoire. Cette approche a été testée avec succès en utilisant différents concepts flous (distance, orientation, taille...), aussi bien dans des cas synthétiques que dans des conditions réelles. Nous étudions également l'intégration de contraintes spatiales floues dans le filtre particulaire, dans le cas multi-objets. Ce type d'information permet de représenter des contraintes complexes entre deux, trois ou plus, objets, telles que les relations « est à gauche de », « est perpendiculaire à », « est aligné avec ». A cette contribution, nous ajoutons celle d'un nouveau modèle de filtre particulaire pour le suivi multi-objets, qui est une extension plus robuste de l'échantillonnage partitionné proposé par J. MacCormick et A. Blake, et qui possède en plus les avantages de l'échantillonnage partitionné à branchements, proposé par ces mêmes auteurs.

Enfin, nous proposons de définir des vraisemblances à modèles multiples. Cette notion est adaptée lorsque l'objet change d'apparence au cours du temps. C'est par exemple le cas avec le suivi de lèvres, qui possèdent des caractéristiques de couleur ou de forme différentes selon l'état de la bouche (« fermée » ou « ouverte »). En définissant la vraisemblance par une loi de mélange dont les poids dépendent de l'objet (et/ou de l'image), il est ainsi possible de pondérer les vraisemblances en considérant l'état dans lequel se trouve l'objet. Les poids peuvent être appris par un classifieur, tel qu'une machine à vecteurs de support (SVM). Ainsi, en utilisant la séparation effectuée par le SVM entre les différents modèles, nous définissons une probabilité d'appartenance, selon une caractéristique choisie, d'un objet à une classe, ce qui définit nos poids.

Toutes ces contributions ont été évaluées sur des séquences réelles, et les résultats montrent l'apport des approches proposées par rapport à d'autres filtres particulaires récents.

Abstract

In this thesis, we are interested in object tracking based on particle filtering approaches in 2D image sequences. Sequential Monte Carlo methods, or particle filters, have known a growing success since the end of the 1990's in the object tracking community, thanks to their ability of adaptation to this kind of problematics, their computational efficiency and their easiness of implementation.

In this context, we propose an original method for the integration of fuzzy information into a particle filter, in order to help the object tracking procedure using spatial information, which may be structural or of shape, or can also correspond to spatial constraints in the multi-object case. The use of spatial information is known to be efficient in numerous image processing fields (segmentation, recognition, ...), but has, to our knowledge, never been used in the particle filter methodological context.

We first establish a general model to manipulate fuzzy spatial information. Each fuzzy relation or shape feature is considered as a linguistic variable, taking a small number of values. The semantic of each value is given by a fuzzy set on the variable domain. Hence, for example, the object orientation may be decomposed according to a finite number of fuzzy sets: "north", "north-west", "east", ... In a similar way, the fuzzy concept "mouth shape" may take values such as "open", "close", and so on.

The defined fuzzy spatial information is introduced in a particle filter using the dynamics of the unknown state, describing the object to track of the system. This approach allows us to add a semantic information to the modeling of the tracking problem. Integrating a fuzzy spatial information in the state space allows better handling the particles in the propagation step. We then exploit the fact that these data do not generally evolve during time: we consider, for example, that the orientation "north-east" must carry on time, but also allowing changes towards a neighbor state, such as values "north" or "east". Unlike classical methods, this kind of information allows us to integrate in the particle filter a synthetized information of the k last estimations. This may improve the tracking robustness against noise. Moreover, in order to predict the relevant fuzzy set, we also use the likelihood to better handle quick changes of the trajectory. This approach has been successfully experimented using different fuzzy concepts (distance, orientation, size...), in synthetical and real cases.

We also propose to integrate fuzzy spatial constraints into the particle filter in the multi-object case. This type of information allows us to represent complex constraints between two, three, or more objects, such as the relations "is at left of", "is perpendiculary of" or "is aligned with". The definition of a new multi-object particle filter is another part of this contribution, and improves the robustness of the partitioned sampling algorithm

proposed by J. MacCormick and A. Blake, and also possesses the same advantages of the branched partitioned sampling, proposed by these authors.

Finally, we propose to define multiple model likelihoods. This notion is adapted when the object appearance evolves during time. It is for example the case with the mouth tracking, which possesses different color and shape features according to the state of the mouth ("close" or "open"). Defining a likelihood by a mixture distribution whose weights depend on the object and the image makes it possible to weight likelihoods considering the state in which the object is. The weights may be learned by a classifier, such as a support vector machine (SVM). Hence, by using the separation carried out by the SVM between the different models, we define a membership probability, according to a chosen feature, of an object to a class, which defines the weights.

All these contributions have been evaluated on real sequences, and the results show the relevance of the proposed approaches compared to other classical particle filters.

Remerciements

Je souhaite remercier en premier lieu mes directrices de thèse, Séverine Dubuisson et Isabelle Bloch, pour m'avoir offert l'opportunité d'effectuer ce doctorat et m'avoir toujours très justement guidé tant sur les aspects techniques que rédactionnels. Je vous remercie également de votre soutien sans faille. La confiance dont vous avez fait preuve m'a aidé à persévérer et à franchir les périodes les plus délicates de ces trois dernières années.

Je remercie Mylène Masson et Wojciech Pieczynski d'avoir accepté de rapporter ce manuscrit, contribuant ainsi à son amélioration. J'exprime toute ma gratitude aux autres membres du jury, Elise Arnaud, Maurice Milgram et Jacques Blanc-Talon, qui ont manifesté de l'intérêt pour ce travail et participé à son évaluation. Mes remerciements vont également à la Direction Générale de l'Armement pour le financement de cette thèse.

Un grand merci à tous les membres du laboratoire TSI de Télécom ParisTech, en particulier ceux du groupe médical, pour tous les bons moments passés en leur présence : Ana, Carolina, David, Emi, d'Jo, Jérémy, Juan-Pablo, Julien, Lazar, Olivier, Patricia, Phong, Vincent, mais aussi les membres d'autres groupes : Aymen, Benoît, Chuck, Fabrice, Gabrielle, Guillaume et Pierre.

Je remercie également les personnes qui sont intervenues dans mon cursus scolaire et m'ont guidé dans mes choix professionnels : mes anciens maîtres de stage (Jérôme G., Fabien et Fawzi), et l'ensemble du LRDE, notamment Jérôme D., Roland, Sylvain, et Théo.

Enfin, je remercie mes proches, qui m'ont aidé souvent sans le savoir, en s'intéressant à ces choses incompréhensibles qui constituent une thèse, ou tout simplement en m'écoutant : ma famille (avec bien sûr une mention spéciale à mes parents), ma bellefamille Paquet, et mes amis (Alex, Amélie, Angélica, Arnaud, Aurore, Cécé, Dédé, Ju, Largif, Pauline, Romain, Ségo, Sido, Tonio et Yoyo). Une dernière pensée particulière pour Nolwenn, qui a fait preuve d'écoute, de compréhension, et qui a su me soutenir et m'encourager dans tous ces moments difficiles qui font la différence. 8_____

Table des matières

Introduction

1	Etat	de l'ar	t	17
	1.1	Métho	odes de Monte-Carlo	18
		1.1.1	Principe de Monte-Carlo	18
		1.1.2	Méthodes non itératives de simulation de lois	19
		1.1.3	Méthodes de Monte-Carlo par chaînes de Markov	23
	1.2	Métho	odes séquentielles de Monte-Carlo	28
		1.2.1	Filtrage particulaire	29
		1.2.2	Rééchantillonnage et nombre efficace de particules	31
		1.2.3	Estimateurs	35
		1.2.4	Evaluation de l'erreur	35
		1.2.5	Fonctions d'importance	37
		1.2.6	Filtre particulaire auxiliaire	38
	1.3	Appli	cations du filtre particulaire en suivi d'objets dans des séquences	
		d'ima	ges	41
		1.3.1	Modélisation du problème	41
		1.3.2	Filtres particulaires à modèles multiples	46
	1.4	Concl	usion	50
2	Мос	délisati	on d'informations spatiales floues	53
	2.1	Inform	nations spatiales floues	54
		2.1.1	Information floue	55
		2.1.2	Fonctions floues	55
		2.1.3	Stratégies de fusion	57
		2.1.4	Ensemble flou des événements possibles	58
2.2 Relations spatiales floues		ons spatiales floues	59	
		2.2.1	Mise en œuvre	60
		2.2.2	Transition	65
	2.3	Inform	nations de formes floues	66
		2.3.1	Mise en œuvre générique	66
		2.3.2	Application à la modélisation de la bouche	68
	2.4	Concl	usion	70

9

3	Inté	gratior	n d'informations spatiales floues dans le filtre particulaire	73
3.1 Introduction d'informations spatiales floues dans un cor			luction d'informations spatiales floues dans un contexte probabiliste	74
	3.2	Introd	luction d'informations spatiales floues dans un filtre particulaire	77
		3.2.1	Modélisation des paramètres dans un filtre particulaire	77
		3.2.2	Modélisation des paramètres dans un contexte flou	78
		3.2.3	Modèle à sauts	79
	3.3	Algor	ithme de suivi	80
	3.4	Fonct	ion d'importance	81
		3.4.1	Modélisation de la fonction d'importance	83
		3.4.2	Complexité et coût de calcul	87
		3.4.3	Conclusion	88
	3.5	Modé	lisation d'informations spatiales floues en pratique	88
		3.5.1	Modélisation des relations spatiales	88
		3.5.2	Modélisation d'informations spatiales générales	89
	3.6	Appro	oximation	90
	3.7	Concl	usion	91
	-			
4	Exp	erimen	tations sur l'integration d'informations spatiales floues dans le filtre	2 02
	pari	Drácor	e ntation dos séguenços do tosts	93
	4.1	A 1 1		94
		4.1.1	Sequences synthetiques	93
		4.1.2		102
	4.0	4.1.3		102
	4.2		Madela flow	107
		4.2.1	Modele flou	108
	1 2	4. <i>Z</i> . <i>Z</i>	Fonction d'Importance	
	4.3	Les re		115
		4.3.1		113
	4.4	4.3.2 L'info	remps de calcul et robustesse des modeles proposes	110
	4.4	L INIO	rmation de formes noues en pratique	124
	4.3	Conci	usion	120
5	Inté	gratior	n de contraintes spatiales floues pour le suivi multi-objets	129
	5.1	Modé	liser des contraintes spatiales floues entre objets	131
		5.1.1	Modélisation des contraintes spatiales floues	131
		5.1.2	Intégration des contraintes spatiales floues dans un contexte pro-	
			babiliste	133
	5.2	Filtrag	ge particulaire pour le suivi multi-objets	134
		5.2.1	Modélisation d'un filtre particulaire multi-objets	134
		5.2.2	Problème de la dimension du vecteur d'état	135
	5.3	L'écha	antillonnage partitionné	137
		5.3.1	Rééchantillonnage pondéré	137
		5.3.2	Echantillonnage partitionné	137
		5.3.3	Discussion	139
	5.4	Echan	tillonnage partitionné à rang	141
	5.5	Expér	iences	146
		5.5.1	Description des séquences	146
		5.5.2	Suivi de piétons	147

		5.5.3	Suivi de fourmis	150
		5.5.4	Suivi de main	152
	5.6	Concl	usion	161
6	Vrai	somble	ances à modèles multiples	165
0	v 1a1	Trava	ux ovietante	165
	0.1	11ava	Multi modolitác	166
		0.1.1	Multi-modalites	100
		6.1.2		100
	$\langle \mathbf{n} \rangle$	6.1.3		169
	6.2	Vraise	emblance a modeles multiples	169
		6.2.1	Iravaux existants	170
		6.2.2	Modèle proposé	171
		6.2.3	Applications	174
	6.3	Machi	ine à vecteurs de support (SVM)	174
		6.3.1	Principe du SVM à deux classes	175
		6.3.2	Extensions à plusieurs classes	177
		6.3.3	Intégration dans une vraisemblance à modèles multiples	177
	6.4	Expér	iences	178
		6.4.1	Protocole de tests	178
		6.4.2	Apprentissage SVM	181
		6.4.3	Résultats préliminaires	185
	6.5	Concl	usion	186
7	Con	clusior	n et perspectives	189
	7.1	Récan	itulatif des contributions	189
	7.1	711	Intégration d'informations floues pour le suivi d'obiet	189
		712	Intégration d'informations floues pour le suivi d'objet multiples	190
		713	Ittilisation de plusieurs modèles d'apparence pour le suivi d'objet	191
		714	Bibliothèque générique de filtre particulaire pour le suivi d'objet	192
	72	Porch	octivos	102
	1.2	721	Intégration d'informations flouos pour le suivi d'obiet	102
		7.2.1	Intégration d'informations floues pour le suivi d'objet	102
		7.2.2	Integration di informations noues pour le suivi d'objets indupies.	193
		7.2.3	Autros novementivos	193
	7 2	7.2.4 D. 1.1:		194
	7.3	Public		194
Α	Con	npléme	ents au chapitre 2 : preuve de la proposition 2.1	197
в	Con	nnléme	ents au chapitre 4 · expériences supplémentaires	201
D	R 1	Influe	nce des paramètres : formes des fonctions d'appartenance et para-	201
	D.1	mètre		201
	в 2	Ságua	nce synthétique : modèle dynamique polaire	201
	D.2 B 3	Séque	nce synthetique. modele dynamique polarie	201 204
	0.0	Jeque	ace reene . volturette	204
С	Bib	liothèq	ue de filtrage particulaire	209
	C.1	Progra	ammation de la bibliothèque	209
	C.2	Exem	ple	211
Bi	bliog	raphie		217
		-		

Introduction

L'un des enjeux en traitement de l'information est de traduire des phénomènes mesurables, que l'on observe à partir d'un appareil d'aquisition, dans un formalisme mathématique. Ce changement de représentation permet de décrire et ainsi comprendre, analyser, prédire le mécanisme physique sous-jacent. Le phénomène mesurable est formellement représenté par une variable d'état, dont la vraie valeur est inconnue. L'objectif consiste à estimer cet état, en fonction d'un ensemble d'observations, qui peuvent être séquentielles, et/ou provenir de plusieurs capteurs.

Dans cette thèse, nous nous intéressons au suivi d'objets dans des séquences d'images, par des techniques bayésiennes, plus particulièrement de filtrage particulaire. La problématique centrale, qui n'avait pas été abordée jusqu'à présent, concerne l'introduction d'informations floues de haut niveau dans de telles techniques. Cela permet de définir des modèles souples génériques et robustes, dans le sens où l'information modélisée s'adapte facilement à n'importe quel type de suivi, en proposant un cadre formel unifié d'informations spatiales floues de différents types.

Inférence bayésienne

De par leur flexibilité en termes de modélisation, les techniques d'inférence bayésiennes permettent, de manière générique, d'estimer la valeur de la variable d'état en fonction d'un ensemble d'observations, et cela dans un cadre probabiliste. La modélisation probabiliste permet d'intégrer une notion d'incertitude sur la valeur du phénomène que l'on cherche à estimer. Lorsque celui-ci évolue au cours du temps, et que l'on collecte ainsi des observations séquentielles, indicées par le temps, il devient possible de modéliser l'évolution du phénomème par un processus stochastique. Dans ce contexte, plusieurs types de traitements émergent :

- la prédiction, qui cherche à estimer l'état à un instant futur selon des observations passées et présentes;
- le lissage, qui consiste à estimer l'état en fonction des observations passées, présentes, et futures;
- et enfin le filtrage, qui a pour objectif d'estimer l'état en fonction des observations passées et présentes.

C'est dans cette dernière catégorie que nous plaçons notre cadre de travail, car elle permet d'estimer l'état d'un phénomène « en ligne », c'est-à-dire en fonction de tout le volume de données qu'il est possible d'aquérir jusqu'à un instant donné.

Formellement, l'objectif de la loi de filtrage consiste à estimer l'état à un instant t, conditionnellement aux mesures, ou observations, passées et courantes, c'est-à-dire du temps 1 au temps t. Deux mécanismes restent alors à définir. Le premier décrit l'évolution de l'état inconnu du système entre deux instants successifs (on parle alors de probabilité de transition), tandis que le second relie les observations courantes à l'état courant, dont la densité de probabilité associée s'appelle vraisemblance. Lorsque ces deux fonctions sont linéaires, et de composantes aléatoires gaussiennes, on dit que le système est linéaire et gaussien. Les lois alors mises en œuvre étant gaussiennes, l'estimation exacte de la loi de filtrage peut se faire de manière analytique, en propageant séquentiellement dans le système les deux premiers moments de cette loi. Les méthodes les plus répandues sont le filtre de Kalman et ses variantes [Bar-Shalom, 1987]. Cependant, dans un cadre général, les densités de probabilité d'évolution et de vraisemblance peuvent être quelconques, et le système non linéaire. Dans ce contexte, les méthodes déterministes de type filtrage de Kalman étendu, sans parfum, et les méthodes à grilles résistent mal respectivement aux fortes non linéarités et aux dimensions de l'espace d'état élevées [Gelb, 1974; Chen, 2003; Kramer et Sorenson, 1988]. Les méthodes séquentielles de Monte-Carlo, encore appelées filtres particulaires [Handschin, 1970; Gordon et al., 1993; Kitagawa, 1996], ne souffrent pas de ces défauts, et proposent d'approcher la loi *a posteriori* de filtrage par une somme finie et pondérée de lois de Dirac, centrées sur des échantillons, appelés particules.

Les méthodes séquentielles de Monte-Carlo en suivi d'objets dans des séquences d'images

En analyse d'images, et plus particulièrement en suivi d'objets dans des séquences d'images 2*D*, les filtres particulaires ont connu un franc succès depuis la seconde moitié des années 1990 [Doucet et al., 2001a]. Leur effacité, leur généricité, leur mise en œuvre simple, leur adaptation naturelle à ce type de problématique et enfin l'évolution technologique des ordinateurs sont les principaux facteurs de cette (subite) notoriété. En suivi d'objets, l'évolution traduit le changement d'état de l'objet entre deux instants successifs, et la vraisemblance l'adéquation entre l'état prédit de l'objet et les données issues de l'image.

En pratique, le manque d'informations sur le modèle d'évolution de l'objet peut limiter les performances du filtre de suivi, ou même le mettre en échec. Nous proposons de remédier à cela en introduisant des informations spatiales de plus haut niveau, qui décrivent l'évolution d'un objet dans le temps. Ces informations peuvent être structurelles, par exemple pour caractériser les positions relatives d'un objet pendant son déplacement, ou des informations telles que des formes. Ces informations sont souvent imprécises et nous proposons donc des modèles flous. En effet, la notion d'information floue permet de traduire une connaissance imprécise, ou vague, d'un phénomène donné, et donc ici de représenter de manière souple l'évolution d'un objet, ce qui n'avait pas, à notre connaissance, été proposé sous cette forme dans un filtre particulaire jusque là. L'approche floue que nous proposons permet en outre de définir une loi de simulation qui lui est dédiée, ce qui améliore les performances du filtre.

Une autre problématique classique consiste à utiliser le filtre particulaire pour suivre plusieurs objets. De nombreux travaux ont été proposés à ce sujet, et nous élaborerons une amélioration d'un filtre déjà existant [MacCormick et Blake, 2000; MacCormick et Isard, 2000]. Nous intégrerons également des contraintes spatiales floues, qui permettent

de définir les relations spatiales entre les objets, et ainsi de restreindre l'espace de leurs configurations possibles.

Lorsque plusieurs observations sont disponibles à un temps *t*, qu'elles soient issues de plusieurs capteurs ou d'un seul d'entre eux, il s'agit d'utiliser au mieux ces différentes modalités, qui peuvent être complémentaires ou redondantes, afin de fournir une estimation de l'état la plus proche possible de la réalité. Là encore, la littérature à ce sujet regorge de contributions. Ici nous nous plaçons dans le cadre d'objets dont l'apparence évolue au cours du temps, et peut être caractérisée par des informations dites multi-modales (couleur, forme, etc.). Dans cette optique, nous proposons de définir des modèles multiples de l'apparence d'un objet, dont le poids de chaque composante est automatiquement estimé par un classifieur SVM.

Organisation du manuscrit

Ce manuscrit de thèse est composé de six chapitres, incluant un chapitre d'état de l'art, et cinq de contributions. Un dernier chapitre conclut ce travail et ouvre quelques perspectives.

Dans le chapitre 1, nous proposons de passer brièvement en revue différentes méthodes de Monte-Carlo pour l'approximation de lois. Elles peuvent être de nature non itératives, itératives (les méthodes de Monte-Carlo par chaîne de Markov), ou séquentielles (filtre particulaire). Une étude de l'utilisation des méthodes de Monte-Carlo séquentielles dans une problématique de suivi d'objets sera également proposée.

Dans le chapitre 2, nous modélisons formellement les informations spatiales floues, qui peuvent être des relations spatiales floues ou encore des informations telles que des formes floues.

Le chapitre 3 concerne l'intégration des informations spatiales floues dans un cadre probabiliste, puis dans un filtre particulaire. Nous proposons également de définir une fonction de simulation, encore appelée fonction d'importance, qui tire profit de la modélisation floue proposée.

Les expériences correspondant à ces deux derniers chapitres seront présentées dans le chapitre 4, qui décrit une étude du comportement de notre méthode par rapport à ses paramètres, avant de la confronter à plusieurs scénarios de suivi, et en comparant ses résultats à ceux obtenus par d'autres filtres particulaires classiques.

Le chapitre 5 traitre du filtre particulaire multi-objets. Sa mise en œuvre lorsque le nombre d'objets est grand est discutée, et nous proposons à ce sujet une amélioration d'un filtre existant, qui utilise une procédure d'échantillonnage partitionné. Nous intégrons également des relations spatiales floues entre objets, afin de contraindre l'estimation à des configurations spatialement cohérentes.

Le chapitre 6 est dédié à l'intégration de plusieurs apparences (observations multiples) dans le modèle de vraisemblance. Les poids associés à chacune des vraisemblances peuvent être estimés par un partitionnement de données (*data clustering* en anglais), ou un classifieur, comme nous le proposons avec une machine à vecteurs de support (SVM). _____

Chapitre 1

Etat de l'art

Sommaire

1	1.1 Méthodes de Monte-Carlo			18	
		1.1.1	Principe	e de Monte-Carlo	18
		1.1.2	Méthod	es non itératives de simulation de lois	19
			1.1.2.1	Méthode de la transformée inverse	19
			1.1.2.2	Algorithme d'acceptation / rejet	20
			1.1.2.3	Échantillonnage par importance	21
		1.1.3	Méthod	es de Monte-Carlo par chaînes de Markov	23
			1.1.3.1	Chaînes de Markov pour la simulation de Monte-Carlo	24
			1.1.3.2	Algorithme de Metropolis-Hastings	25
			1.1.3.3	Echantillonneur de Gibbs	27
1	.2	Métho	odes séqu	entielles de Monte-Carlo	28
		1.2.1	Filtrage	particulaire	29
		1.2.2	Rééchan	tillonnage et nombre efficace de particules	31
		1.2.3	Estimate	eurs	35
		1.2.4	Evaluati	on de l'erreur	35
		1.2.5	Fonctior	ns d'importance	37
		1.2.6	Filtre pa	rticulaire auxiliaire	38
1	.3	Appli	cations du	a filtre particulaire en suivi d'objets dans des séquences	
		d'ima	ges		41
		1.3.1	Modélis	ation du problème	41
			1.3.1.1	Modélisation de la variable d'état	41
			1.3.1.2	Fonctions de transition / proposition	43
			1.3.1.3	Fonctions de vraisemblance	43
			1.3.1.4	Discussion	45
		1.3.2	Filtres p	articulaires à modèles multiples	46
			1.3.2.1	Filtre particulaire à modèles multiples	47
			1.3.2.2	Filtre particulaire à modèles multiples auxiliaire	48
1	4	Concl	usion		50

En préambule, faisons quelques remarques sur le système de notation adopté dans ce document.

Notation 1.1 Par mesure de simplicité, on adoptera dans ce document la convention décrite dans [Doucet et al., 2001a] et largement utilisée dans le domaine. Ainsi, \mathbf{x} dénotera à la fois une variable aléatoire et sa réalisation. On exprimera les distributions de probabilité continues par $\mathbb{P}(d\mathbf{x})$ au lieu de $\mathbb{P}(\mathbf{X} \in d\mathbf{x})$ et les distributions discrètes par $\mathbb{P}(\mathbf{x})$ au lieu de $\mathbb{P}(\mathbf{X} = \mathbf{x})$, avec $d\mathbf{x}$ un ensemble de support infinitésimal de valeur centrale \mathbf{x} . Si ces distributions admettent des densités par rapport à la mesure de référence, celles-ci seront désignées par $p(\mathbf{x})$.

Notation 1.2 Sauf mention explicite contraire, un échantillon désigne une seule réalisation d'une variable aléatoire \mathbf{x} . Dans ce document, il sera noté $\mathbf{x}^{(n)}$ comme le *n*-ième échantillon généré par simulation. Dans le cadre du filtre particulaire, un tel échantillon est appelé particule.

1.1 Méthodes de Monte-Carlo

On appelle méthode de Monte-Carlo un procédé d'estimation de valeurs numériques par simulation aléatoire. Plus le nombre de simulations est important, plus la justesse de l'estimation fournie est fine. Cette technique a pour la première fois été employée en 1733 par Georges-Louis Leclerc de Buffon, dans le cadre d'une expérience visant à approcher le nombre π . Il faudra attendre 1949 pour qu'un premier écrit soit publié sur le sujet [Metropolis et Ulam, 1949], et que, par le même coup, son nom soit proposé, en référence à la nature probabiliste des jeux de hasard notamment pratiqués au casino de Monte-Carlo. Dès lors, son champ d'application n'a cessé de croître, en passant par des domaines aussi variés que ceux de la finance, la biologie, l'ingénierie, la chimie, la physique, l'économie, la statistique, etc. Des moyens de calculs informatiques ont depuis les années 1980 permis de mettre en œuvre ces techniques d'estimation par simulation distribuée. En pratique, elles permettent de résoudre un certain nombre de problèmes, dont le calcul d'intégrales dans de grandes dimensions, la résolution de systèmes linéaires, de problèmes d'optimisation, etc. Nous présentons ici de manière succincte les principaux résultats connus, de manière à passer rapidement à leur utilisation en pratique.

1.1.1 Principe de Monte-Carlo

L'objectif d'une méthode de Monte-Carlo est d'approcher une intégrale du type :

$$\mathcal{I}(\phi) = \mathbb{E}_{p(\mathbf{x})}[\phi(\mathbf{x})] = \int_{\mathcal{X}} \phi(\mathbf{x}) \, p(\mathbf{x}) \, d\mathbf{x}$$
(1.1)

pour toute fonction $\phi(.)$ continue bornée sur \mathcal{X} . Lorsque la densité de probabilité $p(\mathbf{x})$ ou la fonction $\phi(.)$ sont de formes trop complexes, le calcul analytique de cette intégrale n'est pas envisageable. Quant au calcul déterministe sur une grille, il n'est clairement pas adapté lorsque la dimension de \mathcal{X} est grande. C'est pourquoi il est nécessaire de mettre en œuvre une méthode d'approximation, qui soit, si possible, indépendante de la dimension de l'intégrale. Les méthodes de Monte-Carlo ont cette propriété, et en considérant un ensemble d'échantillons $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ indépendants et identiquement distribués selon p(.), l'approximation de l'intégrale s'écrit :

$$\mathcal{I}_{N}(\phi) = \int_{\mathcal{X}} \phi(\mathbf{x}) P_{N}(d\mathbf{x})$$
$$= \frac{1}{N} \sum_{n=1}^{N} \phi(\mathbf{x}^{(n)})$$
(1.2)

avec $P_N(d\mathbf{x})$ l'estimation empirique de la distribution $\mathbb{P}(d\mathbf{x})$:

$$P_N(d\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{x}^{(n)}}(d\mathbf{x})$$
(1.3)

où $\delta_x(.)$ est la mesure de Dirac au point x. La loi des grands nombres nous assure une convergence presque sûre de l'approximation \mathcal{I}_N :

$$\mathcal{I}_{N}(\phi) \xrightarrow[N \to +\infty]{\text{p.s.}} \mathcal{I}(\phi) \tag{1.4}$$

L'estimateur obtenu est sans biais, et, si $\operatorname{var}_{p(\mathbf{x})}[\phi(\mathbf{x})] < +\infty$, de variance proportionnelle à 1/N. D'après le théorème central limite :

$$\sqrt{N}[\mathcal{I}_N(\phi) - \mathcal{I}(\phi)] \underset{N \to +\infty}{\Longrightarrow} \mathcal{N}\left(0, \operatorname{var}_{p(\mathbf{x})}[\phi(\mathbf{x})]\right)$$
(1.5)

où \implies représente la convergence en distribution. Ainsi une précision M fois plus grande nécessite un échantillon M^2 fois plus grand. A la différence des méthodes d'intégration numérique, il apparaît que la convergence de l'estimé de Monte-Carlo est indépendante de la dimension de l'intégrale, c'est précisément ce qui rend les méthodes de Monte-Carlo si attractives. Malheureusement, en pratique, à part dans des cas spécifiques (loi discrète, loi dont la fonction de répartition est inversible, etc.), échantillonner selon la densité p(.) n'est pas une chose aisée, et fait donc l'objet des sections suivantes.

1.1.2 Méthodes non itératives de simulation de lois

Nous considérons être en possession d'un générateur de nombres pseudo-aléatoires selon la loi uniforme $\mathcal{U}[0,1]$. Nous présentons dans cette section différentes méthodes d'échantillonnage non itératives.

1.1.2.1 Méthode de la transformée inverse

Lorsque nous sommes en possession d'une loi de densité de probabilité dont l'expression analytique est connue, et dont la fonction de répartition F(.) est inversible (F réalise une bijection de $\Omega \rightarrow [0, 1]$), alors générer un échantillon selon ladite loi est chose facile.

Soit deux variables aléatoires $\mathbf{x} \sim \mathbb{P}(.)$ et $\mathbf{u} \sim \mathcal{U}[0,1]$. Afin de générer un échantillon selon la loi cible $\mathbb{P}(.)$, l'objectif consiste à trouver une fonction $h : [0,1] \rightarrow \Omega$ telle que $\mathbf{x} = h(\mathbf{u})$. On a :

$$p(\mathbf{x}) = p(\mathbf{u}) \left| \frac{d\mathbf{u}}{d\mathbf{x}} \right|$$

En intégrant, on obtient

$$F(\mathbf{x}) = \int_{-\infty}^{\mathbf{x}} p(\mathbf{x}') \, d\mathbf{x}' = \int_{-\infty}^{\mathbf{u}} p(\mathbf{u}') \, d\mathbf{u}' = \mathbf{u}$$

d'où $\mathbf{x} = F^{-1}(\mathbf{u})$ et $h = F^{-1}$. Le résultat s'interprète de la manière suivante : pour toute variable aléatoire \mathbf{x} dont la fonction de répartition est inversible, $\mathbf{u} = F(\mathbf{x})$ suit une loi uniforme sur [0, 1], ou encore, la variable aléatoire $F^{-1}(\mathbf{u})$ a la même loi que \mathbf{x} . Le procédé de la méthode par transformée inverse est présentée dans le cas continu (algorithme 1.1) et dans le cas discret (algorithme 1.2).

Algorithme 1.1: Simulation par transformée inverse dans le cas d'une loi continue

Entrée : Fonction de répartition F inversible Sortie : $\mathbf{x} \sim \mathbb{P}(.)$ début $| \mathbf{u} \sim \mathcal{U}[0, 1]$ retourner $F^{-1}(\mathbf{u})$

Algorithme 1.2: Simulation par transformée inverse dans le cas d'une loi discrète

$$\begin{split} \mathbf{Entrée} &: \mathrm{Loi} \ \mathrm{discrète} \ \mathbb{P}(\mathbf{x}) = \sum_{n=1}^{N} \delta_{\mathbf{x}^{(n)}} p(\mathbf{x}) \\ \mathbf{Sortie} &: \mathbf{x} \sim \mathbb{P}(.) \\ \mathbf{début} \\ & \mathbf{u} \sim \mathcal{U}[0,1] \\ & i = 0 \\ & \mathbf{tant} \ \mathbf{que} \sum_{n=0}^{i} p(\mathbf{x}^{(n)}) < u \ \mathbf{faire} \\ & \ \ \mathbf{t} \leftarrow i+1 \\ & \ \mathbf{retourner} \ \mathbf{x}^{(i)} \end{split}$$

1.1.2.2 Algorithme d'acceptation / rejet

Dans bien des cas, l'expression analytique de la densité de probabilité $p(\mathbf{x})$ est inconnue. En admettant que l'on sache d'une part évaluer la quantité $\tilde{p}(\mathbf{x})$ telle que $p(\mathbf{x}) = \tilde{p}(\mathbf{x})/Z_p$ avec Z_p inconnu, et d'autre part générer un échantillon selon une fonction de proposition, $q(\mathbf{x})$ et telle que

$$\exists k \in \mathbb{R} \text{ tel que, } \forall \mathbf{x} \in \mathcal{X}, kq(\mathbf{x}) \geq \tilde{p}(\mathbf{x}),$$

alors la réalisation x est acceptée avec la probabilité $\tilde{p}(\mathbf{x})/[kq(\mathbf{x})]$.

La pertinence du choix de k influe sur le taux d'acceptation de la méthode. En remarquant que la probabilité d'acceptation d'un échantillon est directement liée au choix de k, il convient de choisir k aussi petit que possible. En effet, cette probabilité correspond au rapport des aires des fonctions $\tilde{p}(\mathbf{x})$ et $kq(\mathbf{x})$:

$$\mathbb{E}_{q}[accept(\mathbf{x})] = \mathbb{E}_{q}\left[\frac{\tilde{p}(\mathbf{x})}{kq(\mathbf{x})}\right]$$
$$= \int_{\mathcal{X}} \frac{\tilde{p}(\mathbf{x})}{kq(\mathbf{x})} q(\mathbf{x}) d\mathbf{x}$$
$$= \frac{1}{k} \int_{\mathcal{X}} \tilde{p}(\mathbf{x}) d\mathbf{x}$$
$$= \frac{Z_{p}}{k}$$

Différentes stratégies peuvent être mises en œuvre pour améliorer le taux d'acceptation. Par exemple, lorsque ce calcul est possible, le meilleur choix de k est défini par $k = \sup_{\mathbf{x}} \tilde{p}(\mathbf{x})/q(\mathbf{x})$. Il est également possible d'avoir recours à des méthodes de construction de fonctions de proposition adaptatives, qui consistent à mettre à jour la fonction lorsqu'un échantillon \mathbf{x} est rejeté, en forçant le passage en $\tilde{p}(\mathbf{x})$ [Bishop, 2007]. La simulation par acceptation / rejet est présentée dans l'alogrithme 1.3 et illustrée dans la figure 1.1.

Algorithme 1.3: Simulation par l'algorithme d'acceptation / rejet

Entrées : Fonction de proposition q(.), $\exists k \in \mathbb{R}$ tel que, $\forall \mathbf{x} \in \Omega, kq(\mathbf{x}) \ge \tilde{p}(\mathbf{x})$ Sortie : $\mathbf{x} \sim \mathbb{P}(.)$ début répéter $| \mathbf{x} \sim q(\mathbf{x}) |$ $| \mathbf{u} \sim \mathcal{U}[0, kq(\mathbf{x})]$ jusqu'à $\mathbf{u} < \tilde{p}(\mathbf{x})$ retourner \mathbf{x}



FIG. 1.1 – Illustration de l'algorithme d'acceptation / rejet

1.1.2.3 Échantillonnage par importance

L'échantillonnage par importance (*Importance Sampling*, IS), ou encore échantillonnage préférentiel, vise à résoudre un autre type de problématique : comment évaluer une intégrale de la forme $\mathbb{E}_{p(\mathbf{x})}[\phi(\mathbf{x})] = \int_{\mathcal{X}} \phi(\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$ lorsque la simulation selon p(.)est impossible?

Une première méthode naïve pour résoudre ce problème consisterait à discrétiser l'espace dans lequel évolue x par une grille uniforme et ainsi approcher $\mathbb{E}_{p(\mathbf{x})}[\phi(\mathbf{x})]$ par :

$$\mathbb{E}_{p(\mathbf{x})}[\phi(\mathbf{x})] \simeq \sum_{n=1}^{N} \phi(\mathbf{x}^{(n)}) \, p(\mathbf{x}^{(n)})$$

Malheureusement, le nombre de termes nécessaire à ce calcul est lié de manière exponentielle à la dimensionnalité de x. De plus, il s'agirait de devoir explorer tout l'espace \mathcal{X} , alors qu'une infime proportion seulement est généralement porteuse d'information. L'idéal serait de pouvoir échantillonner là où $p(\mathbf{x})$ prend des valeurs significatives, ou, mieux encore, là où les valeurs de $p(\mathbf{x}) \phi(\mathbf{x})$ sont élevées.

L'échantillonnage par importance, comme la méthode d'acceptation / rejet, utilise une fonction de proposition $q(\mathbf{x})$, selon laquelle il est facile d'échantillonner. Ne connaissant les densités de probabilité qu'à un facteur constant près, on pose :

$$p(\mathbf{x}) = \tilde{p}(\mathbf{x})/Z_p \text{ avec } Z_p \text{ inconnu}$$
$$q(\mathbf{x}) = \tilde{q}(\mathbf{x})/Z_q \text{ avec } Z_q \text{ inconnu}$$

Ainsi, il est possible d'écrire :

$$\begin{split} \mathbb{E}_{p(\mathbf{x})}[\phi(\mathbf{x})] &= \int_{\mathcal{X}} \phi(\mathbf{x}) \, p(\mathbf{x}) \, d\mathbf{x} \\ &= \frac{Z_q}{Z_p} \int_{\mathcal{X}} \phi(\mathbf{x}) \, \frac{\tilde{p}(\mathbf{x})}{\tilde{q}(\mathbf{x})} \, q(\mathbf{x}) \, d\mathbf{x} \\ &\simeq \frac{Z_q}{Z_p} \frac{1}{N} \sum_{n=1}^N \tilde{w}(\mathbf{x}^{(n)}) \, \phi(\mathbf{x}^{(n)}) \quad \text{avec} \quad \begin{cases} \mathbf{x}^{(n)} \sim q(\mathbf{x}) \\ \\ \tilde{w}(\mathbf{x}^{(n)}) \triangleq \frac{\tilde{p}(\mathbf{x}^{(n)})}{\tilde{q}(\mathbf{x}^{(n)})} \end{cases} \end{split}$$

En remarquant que $Z_p = \int_{\mathcal{X}} \tilde{p}(\mathbf{x}) d\mathbf{x}$, on obtient :

$$\begin{aligned} \frac{Z_p}{Z_q} &= \frac{1}{Z_q} \int_{\mathcal{X}} \tilde{p}(\mathbf{x}) \, d\mathbf{x} \\ &= \int_{\mathcal{X}} \frac{\tilde{p}(\mathbf{x})}{\tilde{q}(\mathbf{x})} \, q(\mathbf{x}) \, d\mathbf{x} \\ &\simeq \frac{1}{N} \sum_{n=1}^N \tilde{w}(\mathbf{x}^{(n)}) \quad \text{avec} \quad \begin{cases} \mathbf{x}^{(n)} \sim q(\mathbf{x}) \\ \tilde{w}(\mathbf{x}^{(n)}) \triangleq \frac{\tilde{p}(\mathbf{x}^{(n)})}{\tilde{q}(\mathbf{x}^{(n)})} \end{cases} \end{aligned}$$

Enfin, nous obtenons l'expression de l'approximation de l'espérance $\mathbb{E}_{p(\mathbf{x})}[\phi(\mathbf{x})]$:

$$\mathbb{E}_{p(\mathbf{x})}[\phi(\mathbf{x})] = \mathbb{E}_{q(\mathbf{x})}\left[\phi(\mathbf{x})\frac{p(\mathbf{x})}{q(\mathbf{x})}\right] \simeq \sum_{n=1}^{N} w^{(n)} \phi(\mathbf{x}^{(n)}) \quad \text{avec} \quad \begin{cases} \mathbf{x}^{(n)} \sim q(\mathbf{x}) \\ w^{(n)} \triangleq \frac{\tilde{w}(\mathbf{x}^{(n)})}{\sum_{m=1}^{N} \tilde{w}(\mathbf{x}^{(m)})} \end{cases}$$
(1.6)

Plusieurs remarques pour conclure. A l'instar de l'algorithme d'acceptation / rejet, le choix de la distribution d'importance q(.) est primordial pour l'estimation de l'espérance 1.1. Le poids $\tilde{w}(\mathbf{x}^{(n)})$ est appelé poids d'importance, tandis que $w^{(n)}$ correspond au poids d'importance normalisé. La version normalisée engendre une ambiguïté quant à la qualité de l'approximation de la loi $\mathbb{P}(d\mathbf{x})$, car dans un cas optimal (où q(.) correspond à p(.)) comme dans le pire cas (les deux densités sont très éloignées), les poids sont équipondérés. Malheureusement, cette méthode nécessite un nombre de simulations exponentiel par rapport à la dimension de \mathbf{x} [MacKay, 1998; Robert et Casella, 2005]. La procédure d'échantillonnage par importance est présentée dans l'algorithme 1.4.

L'échantillonnage par importance avec rééchantillonnage (*Importance Resampling*, IR) consiste à insérer une étape de rééchantillonnage dans l'algorithme précédent. L'intérêt de cette étape est d'acquérir un échantillon simulé selon la loi $P_N(d\mathbf{x})$ (équation 1.3). L'idée est de générer un nouvel échantillon à partir de celui construit dans l'équation 1.6,

en éliminant les simulations de poids faible, et en démultipliant celles de poids fort. L'étape de normalisation suivie du rééchantillonnage brise l'indépendance mutuelle de l'échantillon généré, ainsi les résultats de convergence classique ne peuvent plus être directement appliqués [Douc et Moulines, 2008]. La méthode de rééchantillonnage proposée dans l'algorithme 1.5 est celle d'un rééchantillonnage multinomial. Il existe d'autres méthodes, visant plus particulièrement à réduire la variance des particules dupliquées. Nous reviendrons sur ce point plus tard.

Algorithme 1.4: Estimation par l'algorithme d'échantillonnage par importance

Entrée : Fonction de proposition q(.)Sortie : Approximation de $\mathbb{E}_{p(\mathbf{x})}[\phi(\mathbf{x})]$ début pour n = 1, ..., N faire $\mathbf{x}^{(n)} \sim q(\mathbf{x})$ Calcul des poids d'importance normalisés : $w^{(n)} = \frac{\tilde{p}(\mathbf{x}^{(n)})/\tilde{q}(\mathbf{x}^{(n)})}{\sum_{m} \tilde{p}(\mathbf{x}^{(m)})/\tilde{q}(\mathbf{x}^{(m)})}$ retourner $\sum_{n=1}^{N} w^{(n)} \phi(\mathbf{x}^{(n)})$

Algorithme 1.5: Simulation par l'algorithme d'échantillonnage par importance avec rééchantillonnage multinomial

Entrée : Fonction de proposition q(.)Sortie : Approximation de la loi cible $\mathbb{P}(d\mathbf{x}) \simeq P_N(d\mathbf{x})$ début pour n = 1, ..., N faire $\mathbf{x}^{*(n)} \sim q(\mathbf{x})$ Calcul des poids d'importance normalisés : $w^{*(n)} = \frac{\tilde{p}(\mathbf{x}^{*(n)})/\tilde{q}(\mathbf{x}^{*(n)})}{\sum_m \tilde{p}(\mathbf{x}^{*(m)})/\tilde{q}(\mathbf{x}^{*(m)})}$ Rééchantillonnage : pour n = 1, ..., N faire $\begin{bmatrix} \mathbf{x}^{(n)} \sim \sum_{m=1}^N w^{*(m)} \delta_{\mathbf{x}^{*(m)}}(d\mathbf{x}) \\ w^{(n)} = 1/N \end{bmatrix}$ retourner $\sum_{n=1}^N w^{(n)} \delta_{\mathbf{x}^{(n)}}(d\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{x}^{(n)}}(d\mathbf{x})$

1.1.3 Méthodes de Monte-Carlo par chaînes de Markov

L'objectif de cette section est de donner un condensé des résultats classiques obtenus par les méthodes de simulations de Monte-Carlo par chaînes de Markov. De nombreux ouvrages complets sont accessibles, tels que [Robert et Casella, 2005; Andrieu et al., 2003b] ou encore dans un contexte un peu plus large, mais dans un style très didactique, [Bishop, 2007].

Les méthodes de Monte-Carlo par chaînes de Markov ont la particularité d'être mieux adaptées que les méthodes de Monte-Carlo non itératives classiques dans le cas où la variable à simuler possède une dimensionnalité élevée ([Robert et Casella, 2005]), ce qui permet d'approcher des intégrales multidimensionnelles :

$$\mathbb{E}_{p(\mathbf{x})}[\phi(\mathbf{x})] = \int_{\mathcal{X}} \phi(\mathbf{x}) \, p(\mathbf{x}) \, d\mathbf{x} \quad \text{avec} \quad \mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_M)$$
$$= \int_{\mathcal{X}_1} \dots \int_{\mathcal{X}_M} \phi(\mathbf{x}) \, p(\mathbf{x}) \, d\mathbf{x}_1 \dots d\mathbf{x}_M$$
$$\simeq \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}^{(n)}) \quad \text{avec} \quad \mathbf{x}^{(n)} \sim p(\mathbf{x})$$
(1.7)

Encore une fois, n'ayant connaissance de la densité de probabilité qu'à une constante près, on pose $p(\mathbf{x}) = \tilde{p}(\mathbf{x})/Z_p$.

1.1.3.1 Chaînes de Markov pour la simulation de Monte-Carlo

Définition 1.1 Un processus $(\mathbf{x}_t)_{t \in \mathbb{N}}$ défini sur l'espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$ et à valeurs dans $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ est une chaîne de Markov d'ordre 1 si pour tout $t \in \mathbb{N}$ et tout borélien S :

$$\mathbb{P}(\mathbf{x}_{t+1} \in S | \mathbf{x}_0, \dots, \mathbf{x}_t) = \mathbb{P}(\mathbf{x}_{t+1} \in S | \mathbf{x}_t)$$
(1.8)

La loi d'une chaîne de Markov est donnée par sa loi initiale $p(\mathbf{x}_0)$, et par son noyau de transition $T_t(\mathbf{x}_t, S) \triangleq \mathbb{P}(\mathbf{x}_{t+1} \in S | \mathbf{x}_t)$, qui représente la probabilité de se déplacer de \mathbf{x}_t vers un élement de S.

Nous considérons ici des chaînes de Markov homogènes, c'est-à-dire $T_t(\mathbf{x}_t, S) = T(\mathbf{x}_t, S), \forall t \in \mathbb{N}$. Le principe des méthodes de Monte-Carlo par chaînes de Markov consiste à construire une chaîne de Markov convergeant vers la distribution invariante $\mathbb{P}(.)$, indépendamment des conditions initiales $\mathbf{x}_0 \sim p(\mathbf{x}_0)$. La densité de probabilité $p(\mathbf{x})$ étant connue à un facteur multiplicatif près, il s'agit alors de créer un noyau de transition $T(\mathbf{x}_t, S)$ dont la *n*-ième itération converge vers $p(\mathbf{x})$, pour $n \to +\infty$.

Définition 1.2 *Une chaîne de Markov est de distribution invariante* $\mathbb{P}(.)$ *si son noyau de transition vérifie :*

$$\mathbb{P}(d\mathbf{x}^*) = \int_{\mathcal{X}} T(\mathbf{x}, d\mathbf{x}^*) \, p(\mathbf{x}) \, d\mathbf{x}$$
(1.9)

Supposons maintenant que le noyau de transition s'écrive sous la forme suivante :

$$T(\mathbf{x}, d\mathbf{x}^*) = f(\mathbf{x}, \mathbf{x}^*) d\mathbf{x}^* + r(\mathbf{x}) \delta_x(d\mathbf{x}^*)$$
(1.10)

où f(x,x) = 0, $\int_{\mathcal{X}} f(x,y) dy \leq 1$, et $r(x) \triangleq 1 - \int_{\mathcal{X}} f(x,y) dy$ représente la probabilité que la chaîne reste en x.

Proposition 1.1 Si la fonction f(.,.) satisfait la condition de réversibilité :

$$p(\mathbf{x}) f(\mathbf{x}, \mathbf{x}^*) = p(\mathbf{x}^*) f(\mathbf{x}^*, \mathbf{x})$$
(1.11)

alors p(.) est la densité invariante associée au noyau $T(\mathbf{x}, .)$.

La preuve de la proposition précédente est disponible dans [Tierney, 1994; Chib et Greenberg, 1995].

Nous supposerons vérifiées les hypothèses d'apériodicité et d'irréductibilité de la chaîne de Markov générée, garantissant l'unicité de la distribution invariante (voir [Meyn et Tweedie, 1993] pour plus de détails).

1.1.3.2 Algorithme de Metropolis-Hastings

Comme nous venons de le constater, le principe d'une méthode de Monte-Carlo par chaînes de Markov consiste à poser un noyau de transition selon l'équation 1.10 puis à définir une fonction f(.,.) qui satisfasse les conditions fixées par l'équation 1.11, assurant la convergence de la chaîne vers la distribution invariante 1.9.

Supposons que l'on sache générer un échantillon selon une densité d'importance q(.,.) telle que $\int_{\mathcal{X}} q(.,x) dx = 1$. En posant f(x,y) = q(x,y), les conditions exprimées dans les équations 1.10 et 1.11 ne seront nécessairement pas vérifiées, et mèneront à l'inégalité suivante, pour un x et un x^{*} donnés :

$$p(\mathbf{x}) f(\mathbf{x}, \mathbf{x}^*) > p(\mathbf{x}^*) f(\mathbf{x}^*, \mathbf{x})$$

L'idée consiste alors à rééquilibrer l'inégalité précédente en favorisant les déplacements de x^* vers x et en défavorisant ceux de x vers x^* . Soit $A(x, x^*)$ la probabilité d'acceptation de x vers x^* , on pose alors :

$$f(\mathbf{x}, \mathbf{x}^*) = \begin{cases} q(\mathbf{x}, \mathbf{x}^*) A(\mathbf{x}, \mathbf{x}^*) & \text{si } \mathbf{x}^* \neq \mathbf{x}; \\ 0 & \text{sinon.} \end{cases}$$

Nous souhaitons ensuite fixer $A(\mathbf{x}, \mathbf{x}^*)$ telle que la condition de réversibilité soit satisfaite :

$$p(\mathbf{x}) f(\mathbf{x}, \mathbf{x}^*) = p(\mathbf{x}^*) f(\mathbf{x}^*, \mathbf{x})$$
$$p(\mathbf{x}) q(\mathbf{x}, \mathbf{x}^*) A(\mathbf{x}, \mathbf{x}^*) = p(\mathbf{x}^*) q(\mathbf{x}^*, \mathbf{x}) A(\mathbf{x}^*, \mathbf{x}) \quad \mathbf{x} \neq \mathbf{x}^*$$

En fixant avec une probabilité de 1 les déplacements défavorisés, c'est-à-dire $A(\mathbf{x}^*, \mathbf{x}) = 1$, on obtient l'expression de la probabilité de mouvement :

$$A(\mathbf{x}, \mathbf{x}^*) = \begin{cases} \min\left\{1, \frac{p(\mathbf{x}^*)q(\mathbf{x}^*, \mathbf{x})}{p(\mathbf{x})q(\mathbf{x}, \mathbf{x}^*)}\right\} & \text{si } p(\mathbf{x})q(\mathbf{x}, \mathbf{x}^*) > 0; \\ 1 & \text{sinon.} \end{cases}$$
(1.12)

f(.,.) satisfaisant par construction la condition de réversibilité, on obtient l'expression finale du noyau de transition décrite dans l'équation 1.11, admettant $\mathbb{P}(d\mathbf{x}^*)$ comme distribution invariante. Enfin, p(.) étant présent au numérateur comme au dénominateur dans l'équation 1.12, nous obtenons l'algorithme 1.6, avec $\tau_{fin} + 1$ le nombre maximal d'itérations, τ_{debut} la longueur de la période de transition, et $\lfloor x \rfloor$ la partie entière inférieure de x.

Le choix d'une fonction de proposition q(.,.) efficace se pose alors. Celui fait dans les travaux précurseurs de [Metropolis et al., 1953] consiste à employer une densité symétrique, c'est-à-dire q(x, y) = q(y, x), dont les variables sont liées par un processus linéaire,

Algorithme 1.6: Simulation par l'algorithme de Metropolis-Hastings

Entrées : Fonction de proposition q(.,.), état initial $\mathbf{x}^{(0)}$ Sortie : Approximation de la loi $\mathbb{P}(d\mathbf{x}) \simeq P_N(d\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{x}^{(n)}}(d\mathbf{x})$ début pour $\tau = 0, ..., \tau_{fin}$ faire $\mathbf{x}^* \sim q(\mathbf{x}^{(\tau)},.)$ $\mathbf{u} \sim \mathcal{U}[0,1]$ Calcul de la probabilité d'acceptation : $A(\mathbf{x}^{(\tau)}, \mathbf{x}^*) = \begin{cases} \min \left\{ 1, \frac{\tilde{p}(\mathbf{x}^*)q(\mathbf{x}^*, \mathbf{x}^{(\tau)})}{\tilde{p}(\mathbf{x}^{(\tau)})q(\mathbf{x}^{(\tau)}, \mathbf{x}^*)} \right\} & \text{si} \quad \tilde{p}(\mathbf{x}^{(\tau)})q(\mathbf{x}^{(\tau)}, \mathbf{x}^*) > 0;$ $1 & \text{sinon.} \end{cases}$ \mathbf{sinon} \mathbf{L} Le candidat est rejeté : $\mathbf{x}^{(\tau+1)} = \mathbf{x}^*$ \mathbf{sinon} \mathbf{L} Le candidat est rejeté : $\mathbf{x}^{(\tau+1)} = \mathbf{x}^{(\tau)}$ $\mathbf{retourner} \left\{ \mathbf{x}^{(i)}; i = \lfloor l(\tau_{fin} - \tau_{debut})/(N-1) \rfloor + \tau_{debut}, l \in \{0, ..., N-1\}, i \in \mathbb{N} \right\}$

y = x + z, avec z un bruit blanc. La probabilité d'acceptation s'en trouve simplifiée, et le tout forme l'algorithme de Metropolis :

$$A(\mathbf{x}, \mathbf{x}^*) = \min\left(1, \frac{\tilde{p}(\mathbf{x}^*)}{\tilde{p}(\mathbf{x})}\right)$$

Ce processus est dit de marche aléatoire, et possède l'inconvénient d'explorer l'espace de manière inefficace, à cause du processus symétrique défini ([Bishop, 2007]). Cet effet est illustré sur la figure 1.2, dans laquelle nous avons donc représenté une simulation par l'algorithme de Metropolis-Hastings avec une fonction de proposition symétrique.

Une deuxième possibilité consiste à utiliser une fonction de la forme $q(x, y) = q_2(y)$, les candidats sont donc générés indépendamment des anciens [Hastings, 1970]). La probabilité d'acceptation correspond alors à :

$$A(\mathbf{x}, \mathbf{x}^*) = \min\left(1, \frac{\tilde{p}(\mathbf{x}^*)q_2(\mathbf{x})}{\tilde{p}(\mathbf{x})q_2(\mathbf{x}^*)}\right)$$
$$= \min\left(1, \frac{\tilde{w}(\mathbf{x}^*)}{\tilde{w}(\mathbf{x}^{(\tau)})}\right)$$

Dans un tel cas, le support de la fonction choisie doit impérativement inclure celui de la loi cible.

Dans [Chib et Greenberg, 1994], les auteurs proposent une solution efficace spécifique. Dans le cas où la densité cible peut être décomposée selon $p(x) \propto \varphi(x)h(x)$, avec $\varphi(.)$ une fonction uniformément bornée et h(.) une densité à partir de laquelle on sait simuler, alors en posant q(x, y) = h(y), on obtient la probabilité d'acceptation suivante :

$$A(\mathbf{x}, \mathbf{x}^*) = \min\left(1, \frac{\varphi(\mathbf{x}^*)}{\varphi(\mathbf{x})}\right)$$

D'autres propositions sont disponibles dans la littérature, comme par exemple dans [Tierney, 1994; Chib et Greenberg, 1995].



FIG. 1.2 – Illustration de l'algorithme de Metropolis-Hastings. La loi de proposition (rouge pointillée) représente une loi normale isotrope, tandis que la loi cible, en bleu, correspond à une loi normale anisotrope.

1.1.3.3 Echantillonneur de Gibbs

L'échantillonneur de Gibbs est un cas particulier de Metropolis-Hastings lorsque $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_M)$ est multidimensionnel et que l'on sait échantillonner selon les lois conditionnelles, ce qui a pour avantage de simuler uniquement sur des distributions univariées.

Le principe de l'échantillonneur de Gibbs [Hastings, 1970; Geman et Geman, 1984] repose sur une propriété issue du produit de noyaux de transition. Considérons $x = (x_1, x_2)$ et supposons qu'il existe un noyau de transition $T_1(x_1, dx_1^*|x_2)$ qui, pour un x_2 donné, soit associé à la distribution d'invariance $\mathbb{P}(dx_1^*|x_2)$ (équation 1.2). De même, supposons qu'il existe un noyau de transition $T_2(x_2, dx_2^*|x_1)$ qui soit associé à la distribution d'invariance $\mathbb{P}(dx_2^*|x_1)$. Il apparaît alors que le produit des noyaux $T_1(x_1, dx_1^*|x_2)$ et $T_2(x_2, dx_2^*|x_1)$ admet comme distribution d'invariance la loi $\mathbb{P}(dx_1^*, dx_2^*)$.

Cette propriété remarquable est exploitée dans l'échantillonneur de Gibbs et nous autorise donc à générer des candidats \mathbf{x}_m^* tour à tour, en utilisant comme noyau de transition les lois conditionnelles totales $\mathbb{P}(d\mathbf{x}_m^* | \mathbf{x}_{\setminus m})$, avec $\mathbf{x}_{\setminus m} = {\mathbf{x}_k}_{k=1:M} \setminus {\mathbf{x}_m}$.

En reprenant l'équation 1.12, la probabilité d'acceptation de la *m*-ème composante

$$\hat{\mathbf{x}}_m^* \triangleq (\mathbf{x}_1, \dots, \mathbf{x}_{m-1}, \mathbf{x}_m^*, \mathbf{x}_{m+1}, \dots, \mathbf{x}_M)$$

s'écrit donc, en identifiant $g(\mathbf{x}, \hat{\mathbf{x}}_m^*) = p(\mathbf{x}_m^* | \mathbf{x}_{\backslash m})$ et pour $p(\mathbf{x})p(\mathbf{x}_m^* | \mathbf{x}_{\backslash m}) > 0$:

$$A(\mathbf{x}, \hat{\mathbf{x}}_{m}^{*}) = \min\left\{1, \frac{p(\hat{\mathbf{x}}_{m}^{*})p(\mathbf{x}_{m}|\hat{\mathbf{x}}_{\backslash m}^{*})}{p(\mathbf{x})p(\mathbf{x}_{m}^{*}|\mathbf{x}_{\backslash m})}\right\}$$
$$= \min\left\{1, \frac{p(\mathbf{x}_{m}^{*}|\mathbf{x}_{\backslash m})p(\mathbf{x}_{\backslash m})p(\mathbf{x}_{m}|\mathbf{x}_{\backslash m}^{*})}{p(\mathbf{x}_{m}|\mathbf{x}_{\backslash m}^{*})p(\mathbf{x}_{\backslash m})p(\mathbf{x}_{m}^{*}|\mathbf{x}_{\backslash m})}\right\}$$
$$= 1$$

La procédure d'approximation par l'échantillonneur de Gibbs est donnée dans l'algorithme 1.7.

Algorithme 1.7: Simulation par l'échantillonneur de Gibbs

Entrée : État initial $\mathbf{x}^{(0)} = (\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_M^{(0)})$ Sortie : Approximation de la loi $\mathbb{P}(d\mathbf{x}) \simeq P_N(d\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{x}^{(n)}}(d\mathbf{x})$ début pour $\tau = 0, \dots, \tau_{fin}$ faire $\mathbf{x}_1^{(\tau+1)} \sim p(\mathbf{x}_1 | \mathbf{x}_2^{(\tau)}, \dots, \mathbf{x}_M^{(\tau)})$ $\mathbf{x}_2^{(\tau+1)} \sim p(\mathbf{x}_2 | \mathbf{x}_1^{(\tau+1)}, \mathbf{x}_3^{(\tau)}, \dots, \mathbf{x}_M^{(\tau)})$ \vdots $\mathbf{x}_m^{(\tau+1)} \sim p(\mathbf{x}_m | \mathbf{x}_1^{(\tau+1)}, \dots, \mathbf{x}_{m-1}^{(\tau+1)}, \mathbf{x}_{m+1}^{(\tau)}, \dots, \mathbf{x}_M^{(\tau)})$ \vdots $\mathbf{x}_M^{(\tau+1)} \sim p(\mathbf{x}_M | \mathbf{x}_1^{(\tau+1)}, \dots, \mathbf{x}_{M-1}^{(\tau+1)})$ retourner $\{\mathbf{x}^{(i)}; i = \lfloor l(\tau_{fin} - \tau_{debut})/(N-1) \rfloor + \tau_{debut}, l \in \{1, \dots, N-1\}, i \in \mathbb{N}\}$

1.2 Méthodes séquentielles de Monte-Carlo

Notons $(\mathbf{x}_t)_{t \in \mathbb{N}}$, respectivement $(\mathbf{y}_t)_{t \in \mathbb{N}^*}$, un processus stochastique à valeurs respectivement dans \mathcal{X} et \mathcal{Y} . La variable d'état $(\mathbf{x}_t)_{t \in \mathbb{N}}$, dont la réalisation est inconnue, est donnée par un processus de Markov de distribution initiale $p(\mathbf{x}_0)$ et de transition $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, et est observée par $(\mathbf{y}_t)_{t \in \mathbb{N}^*}$, dit processus de mesure, selon la famille de densités de probabilité $p(\mathbf{y}_t|\mathbf{x}_t), \forall t \in \mathbb{N}^*$. Cela se traduit par le système suivant :

$$\begin{cases} \mathbf{x}_t = f_t(\mathbf{x}_{t-1}, \mathbf{v}_t) \\ \mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{w}_t) \end{cases}$$
(1.13)

où f_t et h_t sont deux fonctions non linéaires, et où \mathbf{v}_t et \mathbf{w}_t sont deux bruits blancs indépendants. Notons que le modèle graphique définissant les relations entre l'ensemble des variables $(\mathbf{x}_t)_{t \in \mathbb{N}}$ et $(\mathbf{y}_t)_{t \in \mathbb{N}^*}$ n'induit pas la définition du système 1.13, qui n'est en fait qu'un cas particulier satisfaisant les hypothèses imposées par le modèle graphique considéré. En d'autres termes, le modèle décrit ici pourrait très bien n'être écrit qu'en considérant les densités de probabilité correspondantes. Sous les hypothèses markoviennes, d'indépendances conditionnelles des observations sachant le processus d'état, et d'indépendance conditionnelle d'une observation des états passés et futurs sachant l'état courant, on obtient :

$$p(\mathbf{x}_{0:t}) = p(\mathbf{x}_0) \prod_{k=1}^{t} p(\mathbf{x}_k | \mathbf{x}_{k-1})$$
(1.14)

$$p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t}) = \prod_{k=1}^{t} p(\mathbf{y}_k|\mathbf{x}_k)$$
(1.15)

D'où l'expression de la densité a posteriori de la trajectoire :

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}) \frac{p(\mathbf{y}_t|\mathbf{x}_t) \, p(\mathbf{x}_t|\mathbf{x}_{t-1})}{\int_{\mathcal{X}} p(\mathbf{y}_t|\mathbf{x}_t) \, p(\mathbf{x}_t'|\mathbf{y}_{1:t-1}) \, d\mathbf{x}_t'}$$
(1.16)

avec $p(\mathbf{x}'_t|\mathbf{y}_{1:t-1}) = \int_{\mathcal{X}} p(\mathbf{x}'_{t-1}|\mathbf{y}_{1:t-1}) p(\mathbf{x}'_t|\mathbf{x}'_{t-1}) d\mathbf{x}'_{t-1}$ la densité dite de prédiction, qui est obtenue par l'équation de Chapman-Kolmogorov.

L'espérance mathématique $\mathbb{E}_{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}[\phi(\mathbf{x}_{0:t})]$ étant généralement incalculable analytiquement, on souhaiterait estimer l'intégrale de Monte-Carlo associée à la distribution *a posteriori* $\mathbb{P}(d\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$:

$$\mathbb{E}_{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}[\phi(\mathbf{x}_{0:t})] = \int_{\mathcal{X}} \phi(\mathbf{x}_{0:t}) p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t} \quad \text{avec} \quad \mathbf{x}_{0:t} = (\mathbf{x}_{0}, ..., \mathbf{x}_{t})$$
$$\simeq \frac{1}{N} \sum_{n=1}^{N} \phi(\mathbf{x}_{0:t}^{(n)}) \quad \text{avec} \quad \mathbf{x}_{0:t}^{(n)} \sim p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$$
(1.17)

Encore une fois, généralement, il est impossible de générer un échantillon selon $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$. L'utilisation d'une méthode par échantillonnage d'importance est alors envisageable (section 1.1.2.3), selon une densité de probabilité $q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, menant à l'approximation suivante :

$$\mathbb{E}_{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}[\phi(\mathbf{x}_{0:t})] \simeq \sum_{n=1}^{N} w_{0:t}^{(n)} \phi(\mathbf{x}_{0:t}^{(n)}) \quad \text{avec} \quad \begin{cases} \mathbf{x}_{0:t}^{(n)} \sim q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \\ w_{0:t}^{(n)} = \frac{\tilde{w}(\mathbf{x}_{0:t}^{(n)})}{\sum_{m=1}^{N} \tilde{w}(\mathbf{x}_{0:t}^{(m)})} \end{cases}$$
(1.18)

et où $\tilde{w}(\mathbf{x}_{0:t}^{(n)})$ correspond au poids d'importance défini par :

$$\tilde{w}(\mathbf{x}_{0:t}^{(n)}) = \frac{p(\mathbf{x}_{0:t}^{(n)} | \mathbf{y}_{1:t})}{q(\mathbf{x}_{0:t}^{(n)} | \mathbf{y}_{1:t})}$$
(1.19)

Bien que techniquement correcte, cette solution n'en reste pas moins difficilement réalisable en pratique, puisqu'elle nécessite de réestimer les poids d'importance de la trajectoire entière lorsqu'une nouvelle observation est disponible.

C'est dans ce cadre qu'interviennent les méthodes séquentielles de Monte-Carlo, communément appelées filtres particulaires, introduisant l'idée d'échantillonnage par importance séquentiel, et menant à une estimation de complexité linéaire par rapport au temps.

Avant de passer à l'étude du filtrage particulaire, il est certainement intéressant de mentionner le fait que le modèle graphique décrit dans cette section peut s'avérer en réalité simpliste, dans la mesure où les hypothèses qui y ont été formulées ne se justifient que par des raisons pratiques (densités de transition et de mesure simples et intuitives). En particulier, nous avons fait l'hypothèse que les variables $(\mathbf{x}_t)_{t\in\mathbb{N}}$ sont décrites par un processus de Markov. Le modèle induit également la markovianité du couple $((\mathbf{x}_t)_{t\in\mathbb{N}}, (\mathbf{y}_t)_{t\in\mathbb{N}^*})$. Des approches plus générales ont été proposées dans [Desbouvries et Pieczynski, 2003a,b; Derrode et Pieczynski, 2004], qui supposent uniquement le couple $((\mathbf{x}_t)_{t\in\mathbb{N}}, (\mathbf{y}_t)_{t\in\mathbb{N}^*})$ de Markov, ce qui permet d'enrichir le modèle. En particulier, la transition s'écrit $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{y}_{t-2})$ et la vraisemblance $p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{y}_{t-1})$. Bien que représentant une perspective intéressante, ce type de modèle n'a pour l'instant été que très peu considéré en pratique, ce qui est certainement dû à la potentielle difficulté à modéliser ces nouvelles densités de transition et de mesure.

1.2.1 Filtrage particulaire

L'idée consiste à recourir à une simulation séquentielle, en estimant la loi *a posteriori* marginale au temps t, $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$, dite loi de filtrage. Sa densité de probabilité s'obtient

en intégrant sur les états passés à partir de l'équation 1.16 :

$$p(\mathbf{x}_{t}|\mathbf{y}_{1:t}) = \int_{\mathcal{X}^{t}} p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) d\mathbf{x}_{0:t-1}$$

$$= \int_{\mathcal{X}^{t}} p(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}) \frac{p(\mathbf{y}_{t}|\mathbf{x}_{t}) p(\mathbf{x}_{t}|\mathbf{x}_{t-1})}{\int_{\mathcal{X}} p(\mathbf{y}_{t}|\mathbf{x}_{t}') p(\mathbf{x}_{t}'|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t}'} d\mathbf{x}_{0:t-1}$$

$$= \frac{p(\mathbf{y}_{t}|\mathbf{x}_{t}) \int_{\mathcal{X}} p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) p(\mathbf{x}_{t}|\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}}{\int_{\mathcal{X}} p(\mathbf{y}_{t}|\mathbf{x}_{t}') p(\mathbf{x}_{t}'|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t}'}$$
(1.20)

avec $\mathcal{X}^t = \underbrace{\mathcal{X} \times \ldots \times \mathcal{X}}_{t \text{ fois}}$ le produit cartésien des t termes \mathcal{X} .

Lorsque le système 1.13 est linéaire avec des bruits additifs gaussiens, le filtre de Kalman [Kalman, 1960; Welch et Bishop, 1995] donne une estimation analytique exacte de la loi de filtrage $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$. Une solution numérique exacte est également possible, via les méthodes par maillage de l'espace d'état [Kushner, 1990], lorsque l'espace d'état est discret, correspondant ainsi à un modèle de Markov caché. Lorsque les hypothèses de linéarité ne sont pas vérifiées, plusieurs extensions du filtre de Kalman existent. Le filtre de Kalman étendu [Tanizaki et Mariano, 1996; Anderson et Moore, 2005] consiste à approcher la loi de filtrage par une loi gaussienne en linéarisant localement, lorsqu'elles sont dérivables, les fonctions f_t et h_t . Un autre filtre, le filtre de Kalman sans parfum (Unscented Kalman filter) [Julier et Uhlmann, 1997], considère un nuage de points déterminé selon l'approximation gaussienne de la loi *a posteriori* (cette stratégie fait partie des méthodes de transformées sans parfum, ce qui donne le nom au filtre), puis propagé selon le système non linéaire décrit, afin de réestimer les paramètres de la loi d'approximation. Cependant, de par leur approximation gaussienne, ces méthodes ne sont pas adaptées lorsque la densité *a posteriori* est de structure complexe, notamment lorsqu'elle est multimodale.

Au contraire, l'approximation particulaire ne souffre pas de ces défauts, car elle effectue une estimation empirique et non paramétrique de la loi de filtrage :

$$\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t}) \simeq P_N(d\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$$
(1.21)

Dans un cadre séquentiel, les réalisations $\{\mathbf{x}_{0:t}^{(n)}\}_{n=1}^N$ sont appelées particules. Il s'agit alors de définir les nouveaux poids d'importance normalisés $\{w_t^{(n)}\}_{n=1}^N$. En considérant une fonction d'importance telle que

$$q(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = q(\mathbf{x}_{0:t-1}|\mathbf{y}_{1:t-1}) q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, \mathbf{y}_{1:t})$$
(1.22)

et en admettant avoir une estimation de la loi de filtrage à l'instant t - 1, l'estimation de l'espérance mathématique de la loi de filtrage s'écrit alors :

$$\mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[\phi(\mathbf{x}_t)] = \int_{\mathcal{X}} \phi(\mathbf{x}_t) \, p(\mathbf{x}_t|\mathbf{y}_{1:t}) \, d\mathbf{x}_t$$

$$= \int_{\mathcal{X}} \phi(\mathbf{x}_t) \frac{p(\mathbf{y}_t|\mathbf{x}_t) \, p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int_{\mathcal{X}} p(\mathbf{y}_t|\mathbf{x}_t') \, p(\mathbf{x}_t'|\mathbf{y}_{1:t-1}) \, d\mathbf{x}_t'} \, d\mathbf{x}_t$$
(1.23)

Or la densité de probabilité de prédiction $p(\mathbf{x}_t | \mathbf{y}_{1:t-1})$ s'approche sous forme particulaire

par :

$$p(\mathbf{x}_{t}|\mathbf{y}_{1:t-1}) = \int_{\mathcal{X}} p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) p(\mathbf{x}_{t}|\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}$$

$$\simeq \int_{\mathcal{X}} \sum_{n=1}^{N} w_{t-1}^{(n)} \frac{p(\mathbf{x}_{t}|\mathbf{x}_{t-1})}{q(\mathbf{x}_{t}|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t})} q(\mathbf{x}_{t}|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t}) \delta_{\mathbf{x}_{t-1}^{(n)}} (d\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}$$

$$= \sum_{n=1}^{N} w_{t-1}^{(n)} \frac{p(\mathbf{x}_{t}|\mathbf{x}_{t-1}^{(n)})}{q(\mathbf{x}_{t}|\mathbf{x}_{0:t-1}^{(n)},\mathbf{y}_{1:t})} q(\mathbf{x}_{t}|\mathbf{x}_{0:t-1}^{(n)},\mathbf{y}_{1:t})$$

$$\simeq \sum_{n=1}^{N} w_{t-1}^{(n)} \frac{p(\mathbf{x}_{t}^{(n)}|\mathbf{x}_{t-1}^{(n)})}{q(\mathbf{x}_{t}^{(n)}|\mathbf{x}_{t-1}^{(n)})} \delta_{\mathbf{x}_{t}^{(n)}} (d\mathbf{x}_{t}) \quad \text{avec} \quad \mathbf{x}_{t}^{(n)} \sim q(\mathbf{x}_{t}|\mathbf{x}_{0:t-1}^{(n)},\mathbf{y}_{1:t}) \quad (1.24)$$

En réinjectant l'estimation de la densité de prédiction obtenue dans l'équation 1.24, l'estimation de la loi de filtrage de l'équation 1.23 s'écrit :

$$\mathbb{E}_{p(\mathbf{x}_{t}|\mathbf{y}_{1:t})}[\phi(\mathbf{x}_{t})] \simeq \int_{\mathcal{X}} \sum_{n=1}^{N} w_{t-1}^{(n)} \phi(\mathbf{x}_{t}) \frac{p(\mathbf{y}_{t}|\mathbf{x}_{t}) p(\mathbf{x}_{t}^{(n)}|\mathbf{x}_{t-1}^{(n)}) \delta_{\mathbf{x}_{t}^{(n)}}(d\mathbf{x}_{t})}{q(\mathbf{x}_{t}^{(n)}|\mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_{1:t}) \int_{\mathcal{X}} p(\mathbf{y}_{t}|\mathbf{x}_{t}^{(n)}) p(\mathbf{x}_{t}^{(1)}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t}^{\prime}} d\mathbf{x}_{t}}$$

$$= \sum_{n=1}^{N} w_{t-1}^{(n)} \phi(\mathbf{x}_{t}^{(n)}) \frac{p(\mathbf{y}_{t}|\mathbf{x}_{t}^{(n)}) p(\mathbf{x}_{t}^{(n)}|\mathbf{x}_{t-1}^{(n)})}{q(\mathbf{x}_{t}^{(n)}|\mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_{1:t}) \int_{\mathcal{X}} p(\mathbf{y}_{t}|\mathbf{x}_{t}^{(n)}) p(\mathbf{x}_{t}^{\prime}|\mathbf{y}_{1:t-1}) d\mathbf{x}_{t}^{\prime}}$$

$$= \sum_{n=1}^{N} w_{t}^{(n)} \phi(\mathbf{x}_{t}^{(n)}) \quad \text{avec} \begin{cases} \mathbf{x}_{t}^{(n)} \sim q(\mathbf{x}_{t}|\mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_{1:t}) \\ w_{t}^{(n)} = \frac{\tilde{w}(\mathbf{x}_{t}^{(n)})}{\sum_{m=1}^{N} \tilde{w}(\mathbf{x}_{t}^{(m)})} \end{cases}$$
(1.25)

et où $\tilde{w}(\mathbf{x}_t^{(n)})$ correspond au poids d'importance séquentiel défini par :

$$\tilde{w}(\mathbf{x}_{t}^{(n)}) = w_{t-1}^{(n)} \frac{p(\mathbf{y}_{t} | \mathbf{x}_{t}^{(n)}) \, p(\mathbf{x}_{t}^{(n)} | \mathbf{x}_{t-1}^{(n)})}{q(\mathbf{x}_{t}^{(n)} | \mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_{1:t})}$$
(1.26)

(...) (...)

Cette écriture donne naissance au filtre particulaire par échantillonnage séquentiel (SIS) présenté dans l'algorithme 1.8, pour lequel l'estimateur considéré est l'espérance de Monte-Carlo, mais ce point sera discuté plus en détail dans la section 1.2.3. L'entrée de l'algorithme requiert de savoir générer des échantillons selon la loi initiale $\mathbb{P}(d\mathbf{x}_0)$. L'algorithme fait ensuite l'hypothèse que nous savons générer des échantillons selon une fonction de proposition $q(\mathbf{x}_t|\mathbf{x}_{0:t-1},\mathbf{y}_{1:t})$, évaluer la transition *a priori* $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ ainsi que la vraisemblance $p(\mathbf{y}_t|\mathbf{x}_t)$ (toutes deux définies par le système 1.13).

1.2.2 Rééchantillonnage et nombre efficace de particules

Le filtre SIS donne malheureusement de mauvais résultats en pratique. La cause en est la variance des poids d'importance var $[\tilde{w}(\mathbf{x}_t)]$ qui augmente exponentiellement avec le temps [Gilks et Spiegelhalter, 1996], faisant que toutes les particules sauf une convergent alors vers un poids nul. Cet effet indésirable, appelé dégénérescence des particules, mène à la divergence du filtre au bout de quelques itérations. Une solution simple pour limiter ce phénomène consiste à rééchantillonner le nuage de particules obtenu (section 1.1.2.3), en dupliquant les particules de poids fort, et en écartant celles de poids faible. Une autre solution consiste à utiliser une bonne fonction de proposition, c'est-à-dire aboutissant à

Algorithme 1.8: Filtre particulaire à échantillonnage pondéré séquentiel (SIS).

Entrée : Approximation initiale $\mathbb{P}(d\mathbf{x}_0) \simeq \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{x}_0^{(n)}}(d\mathbf{x}_0)$ **Sortie** : Approximation P_N de la loi *a posteriori* de la trajectoire $\mathbb{P}(d\mathbf{x}_{0:T}|\mathbf{y}_{1:T})$ début pour t = 1. T faire

• pour
$$n = 1, ..., N$$
 faire
• Générer $\mathbf{x}_{t}^{(n)} \sim q(\mathbf{x}_{t} | \mathbf{x}_{0:t-1}^{(n)}, y_{t})$
• Poser $\mathbf{x}_{0:t}^{(n)} = (\mathbf{x}_{0:t-1}^{(n)}, \mathbf{x}_{t}^{(n)})$
• Calculer les poids d'importance :
 $\tilde{w}(\mathbf{x}_{t}^{(n)}) = w_{t-1}^{(n)} \frac{p(\mathbf{y}_{t} | \mathbf{x}_{t}^{(n)}) p(\mathbf{x}_{t}^{(n)} | \mathbf{x}_{t-1}^{(n)})}{q(\mathbf{x}_{t}^{(n)} | \mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_{1:t})}$
• pour $n = 1, ..., N$ faire
• Calculer les poids d'importance normalisés :
 $w_{t}^{(n)} = \frac{\tilde{w}(\mathbf{x}_{t}^{(n)})}{\sum_{m=1}^{N} \tilde{w}(\mathbf{x}_{t}^{(m)})}$
• Approximation de la loi marginale au temps t , dite loi de filtrage :
 $\mathbb{P}(d\mathbf{x}_{t} | \mathbf{y}_{1:t}) \simeq P_{N}(d\mathbf{x}_{t} | \mathbf{y}_{1:t}) = \sum_{n=1}^{N} w_{t}^{(n)} \delta_{\mathbf{x}_{t}^{(n)}}(d\mathbf{x}_{t})$
• Espérance de Monte-Carlo issue de la loi de filtrage :

$$\mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[\phi(\mathbf{x}_t)] \simeq \sum_{n=1}^N w_t^{(n)} \phi(\mathbf{x}_t^{(n)})$$

retourner $P_N(d\mathbf{x}_{0:T}|\mathbf{y}_{1:T}) = \sum_{n=1}^N w_T^{(n)} \delta_{\mathbf{x}_{0:T}^{(n)}}(d\mathbf{x}_{0:T})$

une variance des poids minimale, que l'on nomme fonction d'importance optimale. Nous reviendrons plus en détail sur cette dernière dans la section 1.2.5.

Parmi les méthodes de régénération du nuage de particules, le rééchantillonnage multinomial [Gordon et al., 1993; Liu et Chen, 1995] est certainement la plus connue, de par sa simplicité et sa complexité limitée, qui reste linéaire en fonction du nombre de particules. Pour ce faire, il suffit de considérer $N_t^{(n)}$, le nombre de fois où la *n*-ième particule sera dupliquée au temps *t*, comme une variable aléatoire de loi multinomiale et de paramètres *N* et $\{w_t^{*(n)}\}_{n=1}^N$. Cela est équivalent à tirer avec remise les particules proportionnellement à leur poids, c'est-à-dire selon la loi discrète $\sum_{n=1}^N w_t^{*(n)} \delta_{\tilde{\mathbf{x}}_t^{(n)}}(d\mathbf{x}_t)$. La méthode correspondante est résumée dans l'algorithme 1.9. La variance de l'échantillon obtenu est alors var $\left[N_t^{(n)}\right] = Nw_t^{*(n)} - \left(w_t^{*(n)}\right)^2$.

D'autres méthodes de rééchantillonnage ont été proposées dans la littérature, impliquant souvent une variance des particules dupliquées inférieure à celle obtenue par le rééchantillonnage multinomial (voir notamment [Liu et Chen, 1998] pour le rééchantillonnage résiduel, [Kitagawa, 1996; Carpenter et al., 1999] pour celui à variance minimale).

Effectuer un rééchantillonnage systématique peut induire plusieurs effets indésirables. D'une part, cela introduit une perte de la diversité du nuage de particules, ce qui peut s'avérer gênant dans la mesure où l'algorithme utilisé est séquentiel. Lorsque $k \ll t$, avec t l'indice du temps courant, la loi de la trajectoire passée au temps $k \mathbb{P}(d\mathbf{x}_{0:k}|\mathbf{y}_{1:t})$ a toutes les chances d'être représentée par quelques particules, ou même par une seule. Cela est dû aux rééchantillonnages successifs, temps après temps, qui dégradent inévitablement l'estimateur de la loi de la trajectoire. Pour en maintenir la précision, cela nécessiterait un nombre exponentiel de particules par rapport au temps. Cet effet indésirable est intrinsèque à une méthode d'approximation séquentielle de Monte-Carlo, et ne peut être évité. Fort heureusement, la loi de filtrage $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$ conserve de bonnes propriétés en termes de convergence, c'est d'ailleurs ce qui caractérise l'attrait de l'approximation de la loi marginale de filtrage par filtre particulaire (notamment, l'erreur de l'approximation reste indépendante du temps, ce qui n'est pas le cas de la loi de la trajectoire, le lecteur peut par exemple se référer à [Doucet et al., 2001a; Doucet et Johansen, 2009] pour plus de détails, ou aux travaux de P. Del Moral [Del Moral, 1998; Del Moral et Guionnet, 1998, 1999]).

D'autre part, le rééchantillonnage systématique conduit à une augmentation de la variance de l'estimateur de Monte-Carlo [Chopin, 2004], ce qui peut détériorer les performances de l'approximation. Cela induit également une dépendance entre les particules, ainsi les théorèmes de convergence classiques ne peuvent plus être appliqués [Del Moral, 2004].

Afin de limiter le phénomème d'appauvrissement du nuage de particules, des techniques de régularisation ont été proposées dans [Hürzeler et Künsch, 1998], et consistent à perturber les échantillons par un noyau de lissage. Cela a pour effet de diminuer la variance du nuage de particules, mais introduit un biais. D'autres méthodes utilisent une simulation par chaîne de Markov de Monte-Carlo pour traiter le nuage de particules obtenu par l'approximation de la loi jointe de la trajectoire [Gilks et Spiegelhalter, 1996; Gilks et Berzuini, 2001]. Récemment, l'échantillonnage par bloc (*Block Sampling* en anglais) a été proposé dans [Doucet et al., 2006], et permet de diminuer le nombre d'étapes de rééchantillonnage de manière encore plus significative, en opérant directement sur la loi jointe.

Cependant, la première précaution afin de limiter le phénomème d'appauvrissement du nuage de particules est d'effectuer l'opération de rééchantillonnage uniquement lorsque cela est nécessaire. Cette précaution n'est toutefois pas suffisante pour maintenir une diversité des échantillons de la trajectoire, et permet juste d'obtenir de bonnes propriétés des échantillons de la loi de filtrage, c'est-à-dire ceux indicés par le temps présent, t [Del Moral et al., 2008]. Rééchantillonner de manière occasionnelle le nuage de particules implique de définir un critère qui permette de déterminer si un rééchantillonnage est nécessaire. A cet effet, les auteurs dans [Fox, 2003; Soto, 2005] proposent une mesure fondée sur la divergence de Kullback-Leibler tandis que l'auteur dans [Lanz, 2007] utilise un critère entropique pour mesurer l'inadéquation entre la loi de filtrage et la fonction de proposition. Détaillons maintenant le critère le plus couramment utilisé, introduit dans [Kong et al., 1994; Liu et Chen, 1995], qui consiste à comparer le nombre efficace de particules $N_{eff}(N)$ nécessaires à la loi de filtrage afin d'obtenir la même qualité d'estimation qu'avec la fonction de proposition utilisée, avec N particules. Soit :

$$N_{eff}(N) = \frac{N}{1+d(p,q)} \tag{1.27}$$

où d(p,q) représente la χ^2 -distance entre la vraie densité de filtrage $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ et la fonction de proposition $q(\mathbf{x}_t|\mathbf{x}_{0:t}, y_t)$, ce qui correspond à la variance des poids $\operatorname{var}_{q(\mathbf{x}_t|\mathbf{x}_{0:t-1}, y_t)}[w(\mathbf{x}_t)]$. En utilisant une approximation de ce dernier terme fondée sur le coefficient de variation des poids, nous obtenons :

$$\hat{N}_{eff}(N) = \left[\sum_{n=1}^{N} \left(w_t^{(n)}\right)^2\right]^{-1}$$
(1.28)

Ainsi, $\hat{N}_{eff}(N) \in [1, N]$ indique le nombre de particules efficaces, et le test consiste simplement à vérifier que cette valeur n'est pas inférieure à un certain seuil $N_{seuil}(N)$ (la valeur $N_{seuil}(N) = 0.75 N$ est généralement utilisée), indiquant qu'il faudrait $N_{seuil}(N)$ particules à la loi de filtrage pour atteindre la même qualité d'approximation. Des approches de sélection automatique du nombre de particules utilisent également le nombre efficace $N_{eff}(N)$, en forçant l'approximation à toujours correspondre à un nombre de particules efficaces constant [Straka et Šimandl, 2005, 2006].

La procédure d'approximation de la loi de filtrage par un filtre particulaire avec échantillonnage pondéré séquentiel et rééchantillonnage (SIR pour *Sequential Importance Resampling* en anglais) est présentée dans l'algorithme 1.10. Notons que cet algorithme propose de calculer l'estimateur (l'espérance de Monte-Carlo) après la procédure de rééchantillonnage. Cette chronologie est considérée par souci de lisibilité de l'algorithme, mais en pratique, l'évaluation de l'estimateur est faite avant l'étape de rééchantillonnage. En effet, celle-ci ajoute de la variance dans le nuage de particules [Casella et al., 1996], ce qui peut détériorer la qualité de l'estimateur.

1.2.3 Estimateurs

Deux estimateurs sont fréquemment utilisés dans le cadre du filtre particulaire. Le premier est l'espérance de x_t :

$$\mathbf{x}_{t}^{MMSE} = \mathbb{E}_{p(\mathbf{x}_{t}|\mathbf{y}_{1:t})}[\mathbf{x}_{t}] = \int_{\mathcal{X}} \mathbf{x}_{t} \, p(\mathbf{x}_{t}|\mathbf{y}_{1:t}) \, d\mathbf{x}_{t}$$
(1.29)

$$\hat{\mathbf{x}}_{t}^{MMSE} = \sum_{n=1}^{N} w_{t}^{(n)} \mathbf{x}_{t}^{(n)}$$
(1.30)

avec \mathbf{x}_t^{MMSE} l'espérance de \mathbf{x}_t selon $p(\mathbf{x}_t | \mathbf{y}_{1:t})$ et $\hat{\mathbf{x}}_t^{MMSE}$ son estimé, que l'on appelle espérance de Monte-Carlo. Cet estimateur est très largement utilisé dans la communauté, en raison du fait qu'il est sans biais, et a la propriété de minimiser l'erreur quadratique moyenne $\mathbb{E}_{p(\mathbf{x}_t | \mathbf{y}_{1:t})} \left[\left(\hat{\mathbf{x}}_t^{MMSE} - \mathbf{x}_t^{MMSE} \right)^2 \right]$ (MMSE). Un second estimateur est le maximum *a posteriori* (MAP), et correspond au mode le plus important de $p(\mathbf{x}_t | \mathbf{y}_{1:t})$:

$$\mathbf{x}_{t}^{MAP} = \operatorname*{argmax}_{\mathbf{x}_{t}} p(\mathbf{x}_{t} | \mathbf{y}_{1:t})$$
(1.31)

$$\hat{\mathbf{x}}_{t}^{MAP} = \operatorname{argmax}_{\mathbf{x}_{t}} \sum_{n=1}^{N} w_{t}^{(n)} \delta_{\mathbf{x}_{t}^{(n)}}(d\mathbf{x}_{t})$$
(1.32)

avec \mathbf{x}_t^{MAP} le maximum *a posteriori* de \mathbf{x}_t selon $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ et $\hat{\mathbf{x}}_t^{MAP}$ son estimé. En suivi d'objets dans des séquences d'images, cet estimateur est parfois utilisé afin de fournir un résultat visuel le plus proche possible de la réalité. Cependant, il n'intègre pas d'information de densité, et il souffre du fait que son estimation au cours du temps est moins stable que l'espérance, c'est pourquoi son rendu temporel peut être saccadé. D'autres estimateurs peuvent être définis, mais restent très marginaux et très spécifiques dans leur utilisation, le lecteur peut par exemple consulter [Chen, 2003] pour une liste plus exhaustive. Dans le cadre de cette thèse, nous ne considérerons que l'espérance (équation 1.29), qui est l'estimateur le plus naturel pour une méthode d'approximation par Monte-Carlo.

1.2.4 Evaluation de l'erreur

L'erreur quadratique moyenne (RMSE) est la méthode d'évaluation la plus répandue. Nous notons $\text{RMSE}(\mathbf{x}_{1:T})$ l'erreur quadratique moyenne de l'état x au cours du temps 1 à T, $\hat{\mathbf{x}}_t$ l'estimation de l'état au temps t, et \mathbf{x}_t^* sa vraie valeur :

$$RMSE(\hat{\mathbf{x}}_{1:T}) = \sqrt{\frac{1}{T} \sum_{t=1}^{T} \|\hat{\mathbf{x}}_t - \mathbf{x}_t^*\|_2^2}$$
(1.33)

De par la nature stochastique du filtrage particulaire, la quantité 1.33 est également aléatoire.En pratique, c'est d'ailleurs la valeur moyenne sur plusieurs réalisations du RMSE

Algorithme 1.10: Filtre particulaire à échantillonnage pondéré séquentiel avec rééchantillonnage (SIR).

Entrée : Approximation initiale $\mathbb{P}(d\mathbf{x}_0) \simeq \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{x}_n^{(n)}}(d\mathbf{x}_0)$ **Sortie** : Approximation P_N de la loi *a posteriori* de la trajectoire $\mathbb{P}(d\mathbf{x}_{0:T}|\mathbf{y}_{1:T})$ début pour $t = 1, \ldots, T$ faire • pour $n = 1, \ldots, N$ faire • Générer $\tilde{\mathbf{x}}_{t}^{(n)} \sim q(\mathbf{x}_{t} | \mathbf{x}_{0:t-1}^{(n)}, y_{t})$ • Poser $\tilde{\mathbf{x}}_{0:t}^{(n)} = (\mathbf{x}_{0:t-1}^{(n)}, \tilde{\mathbf{x}}_{t}^{(n)})$ • Calculer les poids d'importance : $\tilde{w}(\tilde{\mathbf{x}}_{t}^{(n)}) = w_{t-1}^{(n)} \frac{p(\mathbf{y}_{t}|\tilde{\mathbf{x}}_{t}^{(n)}) p(\tilde{\mathbf{x}}_{t}^{(n)}|\mathbf{x}_{t-1}^{(n)})}{q(\tilde{\mathbf{x}}_{t}^{(n)}|\mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{1:t})}$ • pour $n = 1, \ldots, N$ faire • Calculer les poids d'importance normalisés : $w_t^{*(n)} = \frac{\tilde{w}(\tilde{\mathbf{x}}_t^{(n)})}{\sum_{m=1}^N \tilde{w}(\tilde{\mathbf{x}}_t^{(m)})}$ • si $\hat{N}_{eff}(N) < N_{seuil}(N)$ alors • Rééchantillonner $\{\mathbf{x}_{0:t}^{(n)}, 1/N\}_{n=1}^N$ à partir du nuage $\{\tilde{\mathbf{x}}_{0:t}^{(n)}, w_t^{*(n)}\}_{n=1}^N$ selon l'algorithme du rééchantillonnage multinomial 1.9. sinon • Poser $\{\mathbf{x}_{0:t}^{(n)}, w_t^{(n)}\}_{n=1}^N = \{\tilde{\mathbf{x}}_{0:t}^{(n)}, w_t^{*(n)}\}_{n=1}^N$ • Approximation de la loi marginale au temps *t*, dite loi de filtrage : $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t}) \simeq P_N(d\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$ Espérance de Monte-Carlo issue de la loi de filtrage : $\mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[\phi(\mathbf{x}_t)] \simeq \sum_{n=1}^N w_t^{(n)} \phi(\mathbf{x}_t^{(n)})$ retourner $P_N(d\mathbf{x}_{0:T}|\mathbf{y}_{1:T}) = \sum_{n=1}^N w_T^{(n)} \delta_{\mathbf{x}_{0:T}^{(n)}}(d\mathbf{x}_{0:T})$
qui est considérée, et donc son espérance, ce qui fait d'elle une variable aléatoire. En considérant R évaluations du RMSE, que nous notons $\{\text{RMSE}_r\}_{r=1}^R$, définissons l'écart-type du RMSE :

$$\operatorname{STD}(\hat{\mathbf{x}}_{1:T}) = \sqrt{\frac{1}{R} \sum_{r=1}^{R} \left[\operatorname{RMSE}_{r}(\hat{\mathbf{x}}_{1:T}) \right]^{2} - \left[\frac{1}{R} \sum_{r=1}^{R} \operatorname{RMSE}_{r}(\hat{\mathbf{x}}_{1:T}) \right]^{2}}$$
(1.34)

Cette métrique est rarement utilisée, alors qu'elle permet de calculer la dispersion de l'erreur. Dans le cadre de cette thèse, nous considérerons le RMSE ainsi que son écarttype, afin de juger de la validité et la stabilité du résultat obtenu.

1.2.5 Fonctions d'importance

Le choix de la fonction d'importance, ou fonction de proposition, pour la simulation des particules, est de première importance pour garantir une bonne estimation d'une loi cible. Nous énumérons dans cette section les plus connues.

Le choix le plus répandu consiste à utiliser simplement la densité de transition $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. L'algorithme correspondant est appelé Boostrap [Gordon et al., 1993], lorsqu'aucune stratégie de rééchantillonnage n'est mise en œuvre (ce qui correspond à un filtre SIS (algorithme 1.8) en prenant comme fonction d'importance la fonction de transition), ou encore Condensation [Isard et Blake, 1998a], lorsque le rééchantillonnage est effectué de manière systématique au début d'une nouvelle itération (ce qui correspond à un filtre SIR (algorithme 1.10) en prenant comme fonction d'importance la fonction de transition et avec rééchantillonnage systématique). Cela a l'avantage de rendre simple la mise en pratique du filtre, puisque la mise à jour correspond simplement alors à $\tilde{w}(\mathbf{x}_t) = w_{t-1}^{(n)} p(\mathbf{y}_t|\mathbf{x}_t)$, mais n'utilisant pas la dernière observation, conduit à une augmentation de la variance des poids. En pratique, cela se traduit bien souvent par une faible robustesse du filtre, notamment face aux changements de trajectoire. L'idée centrale des fonctions de proposition consiste donc à prendre en compte l'observation courante pour générer les nouvelles particules.

Dans le cadre des méthodes séquentielles de Monte-Carlo, [Doucet et al., 2000b; Arnaud et Mémin, 2007] ont montré que la fonction d'importance $q(\mathbf{x}_t | \mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_{1:t}) =$ $p(\mathbf{x}_t|\mathbf{x}_{0:t-1}^{(n)},\mathbf{y}_t)$ était optimale au sens où elle minimise la variance des poids d'importance $\tilde{w}(\mathbf{x}_t^{(n)})$ conditionnée à $\mathbf{x}_{0:t-1}^{(n)}$ et \mathbf{y}_t ($\operatorname{var}_{p(\mathbf{x}_t | \mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_t)}[\tilde{w}(\mathbf{x}_t)] = 0$). L'utilisation de cette fonction d'importance mène à l'évaluation des poids $\tilde{w}^{(n)} = w_{t-1}^{(n)} p(\mathbf{y}_t | \mathbf{x}_{t-1}^{(n)})$, avec $p(\mathbf{y}_t|\mathbf{x}_{t-1}^{(n)}) = \int_{\mathcal{X}} p(\mathbf{y}_t|\mathbf{x}_t) p(\mathbf{x}_t|\mathbf{x}_{t-1}^{(n)}) d\mathbf{x}_t$. L'utilisation de la densité d'importance optimale permet d'atténuer le phénomène de dégénérescence des particules, ce qui ne signifie pas qu'elle permet de se soutraire à l'étape de rééchantillonnage. Elle a la propriété de minimiser la variance des poids d'importance, mais cette quantité reste dépendante de la variance de $p(\mathbf{y}_t | \mathbf{x}_{t-1}^{(n)})$, aussi une forte variance de cette densité induit une forte variance de l'estimateur, et donc la nécessité de rééchantillonner le nuage de particules [Doucet et Johansen, 2009]. En pratique, l'échantillonnage selon la fonction d'importance optimale et l'évaluation de la vraisemblance ne sont disponibles analytiquement que si le système d'équations 1.13 est linéaire ou partiellement linéaire (cela correspond au cas où g_t est non linéaire et h_t linéaire). En pratique, l'hypothèse de linéarité sur la fonction h_t est souvent trop forte, et il convient donc de définir d'autres types de fonctions de proposition.

La plus connue est certainement celle définie par [Pitt et Shephard, 1999], donnant naissance au filtre particulaire auxiliaire, et sera étudiée dans la section 1.2.6.

Il est également possible d'avoir recours à un filtre de Kalman étendu [Tanizaki et Mariano, 1996; Anderson et Moore, 2005] ou à un filtre de Kalman sans parfum [Julier et Uhlmann, 1997], par particule. Ainsi, la fonction de proposition de chacune d'elles correspond à un pas de Kalman, ce qui permet d'approcher la fonction de proposition optimale par une densité de probabilité gaussienne. Cette stratégie reste cependant coûteuse en termes de calculs, puisque cela requiert de mettre en œuvre un filtre de Kalman par particule, à chaque temps t.

Des fonctions de proposition plus spécifiques peuvent également être utilisées. Par exemple, [Okuma et al., 2004] utilisent un mélange de lois entre la densité de transition classique et une fonction de détection définie par un algorithme d'apprentissage. Une simulation tirant profit d'une estimation du champ de déplacement par flot optique de la scène a été proposée dans [Arnaud et Mémin, 2007], dans un contexte partiellement linéaire, ce qui permet d'extraire la fonction de proposition optimale.

1.2.6 Filtre particulaire auxiliaire

Le filtre particulaire auxiliaire (AUX-PF) introduit par [Pitt et Shephard, 1999] a pour objectif de définir une fonction d'importance qui est conditionnée par l'ensemble des observations, dont la dernière. La simulation se fait en deux étapes.

La première sert à sélectionner les particules qui pourraient obtenir une forte vraisemblance à l'instant courant, en simplifiant le processus de propagation des particules et en calculant les vraisemblances obtenues. La simplification mentionnée n'est pas obligatoire, et sert en fait à mesurer la vraisemblance selon une propagation qui pourrait être vue comme idéale, nous reviendrons sur ce point plus tard dans cette section. Les particules ayant obtenu des poids forts sont dupliquées par une étape de rééchantillonnage multinomial (algorithme 1.9). La première originalité vient du fait que ce sont les particules à t - 1 qui sont repeuplées ici, même si la procédure est effectuée proportionnellement aux poids qui pourraient être obtenus à t. La première étape sert donc à recréer un nuage de particules de l'état t - 1 dont chaque échantillon est susceptible de survivre à l'instant t.

Une fois le nouveau nuage de particules créé, l'algorithme procède à une étape SIR classique, qui toutefois prend en compte la première étape de l'AUX-PF dans la mise à jour des poids en divisant la vraisemblance par la fonction d'appartenance, comme cela a été vu dans le filtre SIR (algorithme 1.10). La seule particularité, c'est que cette fonction d'appartenance étant définie sur le nuage de particules précédant l'étape de rééchantillonnage, il est donc nécessaire de garder la trace des indices des particules qui ont survécu à cette étape. Le filtre particulaire auxiliaire intègre à cet effet une variable aléatoire discrète, dite auxiliaire, k, à valeurs dans $\{1, \ldots, N\}$, qui désigne l'indice de la particule à t - 1. Elle définit ainsi une notion de parenté d'une particule.

Formellement, il s'agit de définir une fonction d'importance $q(\mathbf{x}_t, \mathbf{k}|\mathbf{y}_{1:t})$, avec k la variable discrète qui encode l'information ancestrale de parenté entre les particules. Cette information permet de faire le lien entre les particules définies à t, et l'indice de leurs parents à t-1. Définissons tout d'abord la densité de probabilité $p(\mathbf{x}_t, \mathbf{k}|\mathbf{y}_{1:t})$ [Arulampalam et al., 2002]:

$$p(\mathbf{x}_{t}, \mathbf{k} | \mathbf{y}_{1:t}) \propto p(\mathbf{y}_{t} | \mathbf{x}_{t}) p(\mathbf{x}_{t}, \mathbf{k} | \mathbf{y}_{1:t-1})$$

$$= p(\mathbf{y}_{t} | \mathbf{x}_{t}) p(\mathbf{x}_{t} | \mathbf{k}, \mathbf{y}_{1:t-1}) p(\mathbf{k} | \mathbf{y}_{1:t-1})$$

$$= p(\mathbf{y}_{t} | \mathbf{x}_{t}) p(\mathbf{x}_{t} | \mathbf{x}_{t-1}^{(\mathbf{k})}) w_{t-1}^{(\mathbf{k})}$$
(1.35)

avec $p(\mathbf{x}_t | \mathbf{k}, \mathbf{y}_{1:t-1}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(\mathbf{k})})$ l'évolution de la k-ième particule et $p(\mathbf{k} | \mathbf{y}_{1:t-1}) = w_{t-1}^{(\mathbf{k})}$ le poids à t - 1 de cette dernière. La fonction d'importance $q(\mathbf{x}_t, \mathbf{k} | \mathbf{y}_{1:t})$ utilisée dans le filtre particulaire auxiliaire est définie selon :

$$q(\mathbf{x}_{t}, \mathbf{k} | \mathbf{y}_{1:t}) = q(\mathbf{x}_{t} | \mathbf{k}, \mathbf{y}_{1:t}) q(\mathbf{k} | \mathbf{y}_{1:t})$$

$$\propto p(\mathbf{x}_{t} | \mathbf{x}_{t-1}^{(\mathbf{k})}) p(\mathbf{y}_{t} | \mu_{t}^{(\mathbf{k})}) w_{t-1}^{(\mathbf{k})}$$

$$= p(\mathbf{y}_{t} | \mu_{t}^{(\mathbf{k})}) p(\mathbf{x}_{t} | \mathbf{x}_{t-1}^{(\mathbf{k})}) w_{t-1}^{(\mathbf{k})}$$
(1.36)

dans laquelle nous avons posé $q(\mathbf{x}_t | \mathbf{k}, \mathbf{y}_{1:t}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(\mathbf{k})})$ et $q(\mathbf{k} | \mathbf{y}_{1:t}) \propto p(\mathbf{y}_t | \mu_t^{(\mathbf{k})}) w_{t-1}^{(\mathbf{k})}$. L'unique différence entre les équations 1.35 et 1.36 est l'utilisation de $\mu_t^{(\mathbf{k})}$, qui reste à définir, dans le calcul de la vraisemblance. Cette valeur $\mu_t^{(\mathbf{k})}$ est une caractéristique statistique de \mathbf{x}_t sachant $\mathbf{x}_{t-1}^{(\mathbf{k})}$, telle que son espérance $\mu_t^{(\mathbf{k})} = \mathbb{E}_{p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(\mathbf{k})})}[\mathbf{x}_t]$, ou un échantillon $\mu_t^{(\mathbf{k})} \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(\mathbf{k})})$ (qui rejoint dans ce cas 1.35). En la définissant par l'espérance, cela permet de s'abstraire de la nature stochastique d'une simulation par $p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(\mathbf{k})})$, dans le but de tester le comportement de la particule dans un cas moyen.

La simulation selon la fonction d'importance définie dans l'équation 1.36 permet d'obtenir un ensemble de N couples $\{(\mathbf{x}_t^{(n)}, \mathbf{k}^{(n)})\}_{n=1}^N$, où $\mathbf{k}^{(n)}$ désigne l'indice de la particule à t - 1 qui est liée à la n-ième particule à t. Elle se fait par une procédure d'échantillonnage par importance, en générant tout d'abord $\mu_t^{(\mathbf{k})}$ selon la transition $p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(\mathbf{k})})$, puis en pondérant les échantillons obtenus par $q(\mathbf{k} | \mathbf{y}_{1:t}) \propto p(\mathbf{y}_t | \mu_t^{(\mathbf{k})}) w_{t-1}^{(\mathbf{k})}$. Les couples $\{(\mathbf{x}_t^{(n)}, \mathbf{k}^{(n)})\}_{n=1}^N$ sont finalement obtenus par rééchantillonnage multinomial, ou de manière équivalente, en sélectionnant les indices avec une probabilité proportionnelle à leur poids. Au terme de cette étape, les couples de particules sélectionnées sont injectés dans une étape SIR classique, en les propageant d'abord, grâce à la notion de parenté définie, puis en calculant leur poids d'importance :

$$\widetilde{w}(\mathbf{x}_{t}^{(n)}, \mathbf{k}^{(n)}) = w_{t-1}^{\mathbf{k}^{(n)}} \frac{p(\mathbf{y}_{t} | \mathbf{x}_{t}^{(n)}) p(\mathbf{x}_{t}^{(n)} | \mathbf{x}_{t-1}^{(\mathbf{k}^{(n)})})}{q(\mathbf{x}_{t}^{(n)}, \mathbf{k}^{(n)} | \mathbf{y}_{1:t})} \\
= \frac{p(\mathbf{y}_{t} | \mathbf{x}_{t}^{(n)})}{p(\mathbf{y}_{t} | \mu_{t}^{(\mathbf{k}^{(n)})})}$$
(1.37)

A l'issue de l'étape de mise à jour des poids, les indices $\mathbf{k}^{(n)}$ sont ignorés et seule la loi de filtrage $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$ est retournée. L'approximation de la loi de filtrage par l'introduction d'une variable aléatoire auxiliaire est présentée dans l'algorithme 1.11.

Le succès de cette méthode est lié à son efficacité, sa simplicité de mise en œuvre, son faible surcoût en termes de complexité de calcul (de O(N) on passe à O(2N) pour le processus d'approximation), son aspect générique (aucune restriction sur le système d'équations défini), et la possibilité d'adaptation à des modèles plus complexes, comme Algorithme 1.11: Filtre particulaire auxiliaire (AUX-PF). **Entrée** : Approximation initiale $\mathbb{P}(d\mathbf{x}_0) \simeq \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{x}_0^{(n)}}(d\mathbf{x}_0)$ **Sortie** : Approximation P_N de la loi *a posteriori* de la trajectoire $\mathbb{P}(d\mathbf{x}_{0:T}|\mathbf{y}_{1:T})$ début pour $t = 1, \ldots, T$ faire 1. Première étape : • pour n = 1, ..., N faire | • Calculer μ_t^n • Calculer les poids d'importance : $\tilde{w}(\mu_t^{(n)}) = w_{t-1}^{(n)} p(\mathbf{y}_t | \mu_t^{(n)})$ • pour $n = 1, \ldots, N$ faire · Calculer les poids d'importance normalisés : $w_t^{*(n)} = \frac{\tilde{w}(\mu_t^{(n)})}{\sum_{m=1}^N \tilde{w}(\mu_t^{(m)})}$ • pour $n = 1, \ldots, N$ faire $\circ~$ Générer l'indice $\mathbf{k}^{(n)} \sim \sum_{m=1}^{N} \delta_{m}^{\mathbf{k}} \, w_{t}^{*(m)}$ 2. Etape SIR : • pour $n = 1, \ldots, N$ faire • Générer $\mathbf{x}_t^{(n)} \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(k^{(n)})})$ et poser $\mathbf{x}_{0:t}^{(n)} = (\mathbf{x}_{0:t-1}^{(k^{(n)})}, \mathbf{x}_t^{(n)})$ • Calculer les poids d'importance : $\tilde{w}(\mathbf{x}_t^{(n)}, \mathbf{k}^{(n)}) = \frac{p(\mathbf{y}_t | \mathbf{x}_t^{(n)})}{n(\mathbf{y}_t | u_t^{(\mathbf{k}^{(n)})})}$ • pour $n = 1, \ldots, N$ faire • Calculer les poids d'importance normalisés : $w_t^{(n)} = \frac{\tilde{w}(\mathbf{x}_t^{(n)}, \mathbf{k}^{(n)})}{\sum_{m=1}^N \tilde{w}(\mathbf{x}_t^{(m)}, \mathbf{k}^{(m)})}$ 3. Approximation de la loi marginale au temps *t*, dite loi de filtrage : $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t}) \simeq P_N(d\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$ 4. Espérance de Monte-Carlo issue de la loi de filtrage : $\mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[\phi(\mathbf{x}_t)] \simeq \sum_{t=1}^N w_t^{(n)} \phi(\mathbf{x}_t^{(n)})$ retourner $P_N(d\mathbf{x}_{0:T}|\mathbf{y}_{1:T}) = \sum_{n=1}^N w_T^{(n)} \delta_{\mathbf{x}_{0:T}^{(n)}}(d\mathbf{x}_{0:T})$

les modèles à sauts [Karlsson et Bergman, 2000]). Cependant, il existe des cas où les performances du filtre particulaire auxiliaire peuvent être moins bonnes que celles du filtre classique (obtenu en générant les particules selon la densité de transition), notamment lorsque le bruit de la dynamique est élevé, puisque dans ce cas, l'approximation considérée par le calcul de μ_t ne suffit plus à représenter convenablement la distribution de transition.

1.3 Applications du filtre particulaire en suivi d'objets dans des séquences d'images

1.3.1 Modélisation du problème

Comme nous l'avons vu, le filtre particulaire est une méthode d'estimation des lois marginales $\mathbb{P}(d\mathbf{x}_k|\mathbf{y}_{1:k})$, *forallk* = 1, ..., *T*.La loi $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$ sert à estimer l'état à un certain temps *t*, conditionné aux mesures passées et présentes, que l'on appelle loi *a posteriori* au temps *t*, ou loi de filtrage. La modélisation du problème fait intervenir deux fonctions, une de transition, qui décrit l'évolution de l'état au cours du temps, et l'autre de mesure, qui décrit l'observation mesurée conditionnée à l'état, et dont la densité de probabilité s'appelle vraisemblance.

En traitement d'images, l'application la plus évidente est celle du suivi d'objet dans une séquence d'images. La transition correspond ainsi à la dynamique de l'objet, et la vraisemblance à une observation issue de l'image qui permette d'évaluer la densité de probabilité conditionnellement à un état. Ce n'est toutefois pas la seule application, et le filtre particulaire peut en effet être employé à des fins de segmentation d'objet dans une image donnée. On appelle cela du suivi spatial, pour lequel la transition correspond à l'évolution présumée de la segmentation, et la vraisemblance une fonction de satisfaction sur l'opération de segmentation. Dans ce cadre, nous pouvons par exemple citer le travail de [Pérez et al., 2001] sur le suivi du contour d'un objet, ou ceux de [Florin et al., 2005; Lesage et al., 2008], qui segmentent des objets de formes tubulaires pour la segmentation vasculaire 3*D*.

Dans le cadre de cette thèse, nous nous intéressons uniquement au suivi d'objets dans des séquences d'images. Nous focalisons donc maintenant notre attention sur la modélisation de la variable d'état pour le problème de suivi (section 1.3.1.1) ainsi que des fonctions de transition, de proposition (section 1.3.1.2), et de vraisemblance (section 1.3.1.3) L'objectif de cette section n'est ni de fournir une description détaillée, ni une bibliographie complète de la mise en œuvre des filtres particulaires en suivi d'objets, mais juste de dessiner les grandes lignes des approches répondant à cette problématique. La discussion à propos des difficultés rencontrées dans le filtrage particulaire est proposée dans la section 1.3.1.4. Les points qui nous intéressent particulièrement seront détaillés dans les chapitres suivants.

1.3.1.1 Modélisation de la variable d'état

La modélisation du vecteur aléatoire de l'état est la première étape dans la mise en œuvre d'un filtre de suivi. Celui-ci contient tous les éléments inconnus que l'on cherche à estimer, conditionnellement à un ensemble d'observations. En matière de suivi d'objets, il est possible de décomposer ces informations en cinq classes.

La première correspond aux informations de positions spatiales, telles que la position de l'objet dans le repère considéré, ses dimensions, ou encore son orientation. Ce sont les informations de base de la plupart des méthodes de suivi.

La deuxième contient des informations sur la représentation de l'objet : sa forme, ses dimensions, son échelle. Les formes peuvent être modélisées sans contrainte *a priori*, c'est-à-dire en définissant un ensemble de points de contrôle indépendants, puis en modélisant le contour par une B-spline [Isard et Blake, 1998a], ou par des descripteurs de Fourier [Hue, 2003]. A l'inverse, il est également possible de les restreindre, souvent via une procédure d'apprentissage, comme c'est par exemple le cas avec des formes quelconques dans [Rathi et al., 2007; Chang et al., 2008; Isard et Blake, 1998a], ou en utilisant un modèle *a priori* de forme, comme un rectangle, une ellipse, un modèle tubulaire dans un contexte de suivi 3*D* de vaisseaux [Florin et al., 2005; Lesage et al., 2008], ou encore un modèle de forme complexe dans le cadre du suivi de tête en 3*D* [Dornaika et al., 2004].

La troisième classe définit toutes les informations nécessaires pour caractériser la dynamique de l'objet, telles que la vélocité, l'accélération, le couple angle et distance dans un système polaire, etc. Là encore, ce type d'information est considéré dans la plupart des algorithmes de suivi.

La quatrième classe d'informations, qui n'est qu'assez rarement modélisée, contient les hyperparamètres [Caron et al., 2005, 2007], tels que la matrice de variance-covariance d'une loi normale, ou encore les probabilités d'une loi de transition discrète. Cela permet d'estimer les paramètres, par exemple d'une loi d'évolution, qui évolueraient au cours du temps.

Enfin, la dernière catégorie définit tout ce qui ne peut être mis dans les quatre premières. C'est souvent la classe qui contient les informations les plus intéressantes, dans le sens où elles constituent la spécificité du modèle considéré. Dans le cas où un objet peut être occulté, les auteurs dans [Pérez et Vermaak, 2005; MacCormick et Blake, 2000] modélisent une variable aléatoire binaire de visibilité de l'objet. Lorsque l'objet peut disparaître totalement ou apparaître dans la scène, ceux dans [Vermaak et al., 2005b; Pérez et Vermaak, 2005] modélisent une variable d'existence. Dans le cas où les objets peuvent être de différentes natures, il est possible d'estimer la catégorie de l'objet [Bardet et al., 2009, 2010]. Afin de gérer de manière plus efficace les conditions de luminosité (telles que la gestion des ombres), les auteurs de [Bardet et al., 2010] proposent de modéliser la présence et la position du soleil. Nous y reviendrons par la suite, mais il est également possible de considérer plusieurs dynamiques pour décrire l'évolution de l'objet, ainsi une méthode consiste à estimer quelle dynamique un objet suit à un instant donné [Isard et Blake, 1998b; Kreucher et al., 2004; McGinnity et Irwin, 2000]. Enfin, dans un contexte multi-objets, il faut modéliser l'ensemble des objets [Hue et al., 2002; Avitzour, 1995; Gordon, 1997; Doucet et al., 2002; Isard et MacCormick, 2001; MacCormick et Blake, 2000] pour ne citer que les plus connus, mais nous reviendrons sur ce point dans le chapitre 5 (il est aussi possible de considérer un filtre particulaire par objet [Schulz et al., 2001; Du et Piater, 2005; Ryu et Huber, 2007; Muñoz-Salinas et al., 2008b]). Lorsqu'une incertitude sur l'origine des mesures existe, il est possible de modéliser directement dans le vecteur inconnu une variable d'association entre les observations et les objets [Ikoma et Godsill, 2003; Särkkä et al., 2004]. Lorsque le nombre d'objets est variable, cette quantité peut également être estimée [Isard et MacCormick, 2001; Khan et al., 2005; Smith et al., 2005; Bardet et al., 2009].

1.3.1.2 Fonctions de transition / proposition

La fonction de transition, qui joue le rôle de fonction de proposition dans les filtres Boostrap et Condensation (section 1.2.5), définit la dynamique a priori de l'objet, sans utiliser d'observation pour générer les particules. Les modèles de transition sont nombreux, mais nous pouvons énumérer la dynamique de marche aléatoire, le modèle à vélocité constante, le modèle à accélération constante, le modèle sinusoïdal, le modèle de virages (notamment utilisé dans des applications de robotique), etc. Lorsque l'objet est décrit par plusieurs dynamiques, il convient de définir des modèles multiples. Dans [Isard et Blake, 1998b], les auteurs proposent d'utiliser un modèle d'accélération constante couplé à un modèle de rebond, dans le cadre du suivi d'une balle rebondissante ; dans [Brasnett et al., 2007], les auteurs combinent un modèle à vélocité constante avec une loi uniforme dans l'espace des valeurs, afin d'autoriser des réinitialisations du filtre, lorsque l'objet a été perdu. Des modèles d'interactions multiples (IMM en anglais, pour Interacting Multiple Model) ont également été proposés dans des filtres de Kalman [Kirubarajan et al., 2001; Benoudnine et al., 2009], dans un filtre particulaire [Kreucher et al., 2004; Guo et al., 2008; Wang et al., 2004; Boers et Driessen, 2003; McGinnity et Irwin, 2000]. Toujours dans le même but, mais avec une mise en œuvre différente, des modèles à sauts markoviens ont vu le jour dans des cadres linéaires [Doucet et al., 2001b; Bergman et al., 2001], et non linéaires [Andrieu et al., 2003a; Driessen et Boers, 2005].

Comme nous l'avons vu dans la section 1.2.5, définir une fonction de proposition n'est pas toujours simple, et peut dépendre des hypothèses effectuées sur le système considéré. Dans la majorité des cas, la fonction de proposition correspond à la fonction de transition, ce qui a l'avantage de rendre le filtre de suivi simple à modéliser, d'obtenir des résultats convenables, tout en limitant le nombre de calculs. Toutefois, l'utilisation d'un filtre particulaire auxiliaire améliore sensiblement les résultats (section 1.2.6), et peut être utilisé dans n'importe quelle condition. L'utilisation de la fonction de proposition optimale a été proposée dans un cadre partiellement gaussien dans [Arnaud et Mémin, 2007], en faisant intervenir une approximation gaussienne pour générer les particules dans [Grisetti et al., 2007], ou en effectuant une approximation par simulation dans [Blanco et al., 2008], dans le cadre de la localisation d'un robot. Une fonction de proposition hybride a également été proposée dans [Huang et Djuric, 2004], combinant des informations issues des densités a posteriori et a priori. Une autre modélisation intéressante de la fonction de proposition consiste à la définir par un procédé issu d'une procédure de traitement d'images. Cela est proposé dans [Arnaud et Mémin, 2007] avec l'utilisation d'un flot optique. Dans [Okuma et al., 2004], les auteurs utilisent un algorithme de détection Adaboost pour détecter la présence ou la disparition des objets, ainsi que pour guider la dynamique des particules. Enfin, dans [Pérez et al., 2004], la fonction de proposition se sert de caractéristiques de déplacements et de sons afin d'attirer les particules vers les zones à fortes activités.

1.3.1.3 Fonctions de vraisemblance

La dernière étape, et certainement la plus importante, concerne la modélisation de la vraisemblance, qui est en charge de pondérer les particules en utilisant des observations issues de l'image. Ces observations peuvent correspondre aux données brutes de l'image, comme les valeurs de pixels, ou à des mesures générées par un procédé de traitement d'images. Nous pouvons distinguer deux types de vraisemblances. Premièrement, celles qui n'utilisent pas de modèle de référence, ce qui inclut par exemple l'utilisation d'une sélection automatique de caractéristiques discriminantes [Collins et al., 2005; Wang et al., 2005], d'une carte de saillance [Zhang et al., 2010], de points de Harris dans un contexte de suivi de points [Du et Piater, 2005], d'une segmentation à partir d'informations de mouvement de type flot optique [Hue, 2003], ou encore de gradient pour décrire un contour [Isard et Blake, 1998a; Brasnett et al., 2007]. Deuxièmement, il y a les vraisemblances qui supposent connu un modèle de référence, qui sert de base à une comparaison entre le modèle défini par une particule (l'état candidat) et le modèle de référence. Nous pouvons énumérer les modèles qui utilisent une information de couleur de l'objet [Nummiaro et al., 2002], potentiellement en modélisant en plus le fond de l'image [Pérez et al., 2002]. Ces modèles utilisent une comparaison des histogrammes de couleurs d'un candidat par rapport à un modèle de référence en utilisant une notion de distance entre les deux histogrammes, qui est généralement la distance de Bhattacharyya [Kailath, 1967; Thacker et al., 1997]. Le modèle de référence peut être mis à jour de manière adaptative, comme cela a notamment été proposé dans [Nummiaro et al., 2002; Muñoz-Salinas et al., 2008a]. Des informations de texture peuvent également être intégrées [Klein et al., 2007; Brasnett et al., 2007]. Une autre vraisemblance utilisant des modèles consiste à comparer deux fenêtres centrées, comme cela a été mis en œuvre dans le cadre d'un suivi de points [Arnaud et Mémin, 2007; Arnaud et al., 2005].

L'utilisation de plusieurs vraisemblances fait également l'objet de nombreux travaux, dont nous ne donnons ici qu'un bref aperçu. On cherche alors à prendre en compte plusieurs modalités de données, ou caractéristiques. Ainsi, les auteurs dans [Brasnett et al., 2007] proposent de combiner des informations de couleurs, textures et contours, [Klein et al., 2007; Maggio et al., 2005] de couleurs et textures, [Moreno-Noguer et al., 2008] de couleurs et contours, [Muñoz-Salinas et al., 2008a] de couleurs et d'une carte de distance, [Erdem et al., 2010] de couleurs, de déplacements et de brillance, ou encore [Muñoz-Salinas et al., 2008b] de couleurs, d'un « Mean-Shift » et d'une carte de profondeur. Pour intégrer plusieurs vraisemblances dans un filtre particulaire, il existe plusieurs stratégies. La plus simple, et la plus commune aussi, consiste à considérer une indépendance conditionnelle des observations sachant les états. Ainsi le produit des vraisemblances associées à une seule caractéristique forme la vraisemblance finale.Tout en gardant l'hypothèse d'indépendance conditionnelle, une autre stratégie peut être utilisée. En considérant la vraisemblance totale comme une loi de mélange, les poids représentent la probabilité que la caractéristique correspondante soit la plus pertinente [Hotta, 2006; Xu et Li, 2005]. Mettre une valeur entre 0 et 1 en exposant d'une vraisemblance indique la confiance que l'on a dans une caractéristique [Brasnett et al., 2007; Erdem et al., 2010]. Ou encore, [Maggio et al., 2005] proposent d'utiliser un principe d'incertitude sur la caractéristique via un critère sur la dispersion de la vraisemblance attachée aux particules. L'intérêt de ces trois stratégies est bien sûr d'adapter les valeurs des paramètres de fusion (probabilité, confiance et incertitude) aux données, dans le but de ne pas prendre en compte une vraisemblance si la caractéristique qu'elle représente n'est pas, à un instant donné, pertinente dans l'image. Une dépendance entre les caractéristiques a été proposée dans [Moreno-Noguer et al., 2008], menant à une estimation séquentielle et conditionnelle de la vraisemblance totale.

La problématique multi-capteurs (qui regroupe l'utilisation de plusieurs capteurs de types différents, ou identiques (multi-vues)) peut être dissociée de celle de multi-vraisemblances, puisque, bien que cela ne soit pas toujours le cas, modéliser des mesures issues de différents capteurs peut donner naissance à des modèles de fusion spécifiques, dans lesquels les informations peuvent être contradictoires ou bien complémentaires. Ainsi dans [Muñoz-Salinas et al., 2009], la fusion de plusieurs caméras se fait en utilisant

45

une carte d'occupation au sol et en hauteur, alors que dans [Wang et al., 2007], la fusion correspond plus simplement à un produit des vraisemblances. Les auteurs dans [Vermaak et al., 2001] proposent d'utiliser les données de manière séquentielle, en repeuplant le nuage de particules autant de fois qu'il y a de capteurs. Toujours dans une application multi-capteurs, dans laquelle chacun d'entre eux peut, en fonction de son état de fonctionnement, fournir différentes observations, les auteurs dans [Caron et al., 2007] proposent d'estimer l'indice du régime adéquat associé à chaque capteur. Enfin, des capteurs de différentes natures peuvent être utilisés. Dans [Cui et al., 2008], les auteurs proposent d'utiliser un scanner laser et une vidéo caméra. Dans [Vermaak et al., 2001; Zotkin et al., 2002; Pérez et al., 2004; Vermaak et al., 2001], il s'agit de la fusion de données issues de la vision et du son. Une complémentarité entre caméra visible et thermique infrarouge a été mise en œuvre dans [Péteri et Šiler, 2009; Erdem et al., 2010].

1.3.1.4 Discussion

Insistons un peu plus sur les difficultés liées à la mise en œuvre des filtres particulaires, que nous avons jusqu'ici entre-aperçues. La première est la modélisation de la dynamique. Dans la grande majorité des cas, et cela est particulièrement vrai en suivi, la dynamique exacte de l'objet est inconnue, tout comme le sont ses paramètres. Il est alors fréquent d'utiliser une dynamique simple, qui ne contraigne pas le suivi, telle qu'une marche aléatoire ou un modèle à vélocité constante. Les paramètres qui définissent le bruit du modèle gaussien alors considéré sont choisis de manière à ne pas perdre l'objet, et donc bien souvent surestimés. Cela est fait au prix d'une potentielle mauvaise prédiction des particules, ce qui conduit à une augmentation de la variance des poids des particules, donc à une diminution du nombre de particules efficaces (section 1.2.2), et à un réechantillonnage qui devient alors systématique. Cela limite la diversité du nuage de particules, et diminue de ce fait la robustesse du suivi [Orton et Fitzgerald, 2002]. L'information pauvre qui est utilisée pendant l'étape de transition est généralement compensée par le choix d'une bonne fonction de vraisemblance, bien souvent très « piquée », afin de repeupler le nuage de particules uniquement par des échantillons ayant obtenu de fortes vraisemblances [Arnaud et Mémin, 2007], ce qui conduit également à une dégénérescence du nuage de particules.

La modélisation de la fonction de proposition est souvent primordiale pour assurer de bonnes performances de suivi. Bien que son importance ne soit plus à démontrer, son utilisation en pratique reste, quant à elle, souvent ignorée. La cause en est la difficulté à la définir, c'est-à-dire à décrire une fonction de proposition qui dépende à la fois de l'état à l'instant précédent (ou aux instants précédents), ainsi que de la nouvelle observation. La fonction de proposition optimale, bien qu'étant, comme son nom l'indique, la meilleure fonction de proposition qu'il soit possible de définir, au sens où elle minimise la variance des poids des particules, reste difficilement appliquable en pratique, à cause de ses hypothèses de linéarités partielles ou totales. La solution de définir une fonction de proposition fournie par un procédé de traitement d'images est bonne, mais souvent trop spécifique pour être généralisable. Enfin, la fonction de proposition sans doute la plus utilisée est celle définie dans le filtre particulaire auxiliaire, et permet, de manière générique, d'utiliser la dernière observation et l'état à l'instant précédent (section 1.2.6). Toutefois, cette fonction de proposition n'utilise pas l'ensemble des états passés, c'està-dire la trajectoire de l'état, qui peut être utile lorsqu'une ambiguïté spatio-temporelle locale survient.

L'utilisation de plusieurs vraisemblances, ou plusieurs capteurs, est délicate, puisqu'elle pose la question des capteurs défaillants, des caractéristiques non pertinentes, etc. Dans la littérature, cela est traité par l'introduction d'une quantité qui indique la plausibilité, la confiance, ou la probabilité d'une caractéristique ou d'un capteur (nous définirons ces termes plus précisément dans le chapitre 6). Ces quantités sont bien souvent estimées à chaque temps en utilisant les vraisemblances obtenues par le nuage de particules, ou celles obtenues par l'état estimé, selon un critère bien choisi. Cette stratégie de mise à jour peut être mise en échec, puisqu'en affectant les mêmes poids à toutes les particules, elle facilite la propagation d'erreurs dans le cas où les poids sont erronés. Ainsi, un nuage de particules mal localisé fournit une mauvaise estimation des indices de plausibilités, ou de confiances, ou des probabilités, ce qui entraîne encore plus rapidement l'échec du suivi.

Enfin, la problématique multi-objets est un sujet complexe sur lequel la littérature est particulièrement prolifique (plus de détails et de références à ce sujet seront donnés dans le chapitre 5). Premièrement, le problème d'association des mesures aux objets, qui est inhérent à une modélisation multi-objets d'un algorithme de suivi, est généralement complexe et mène à des algorithmes qui ne le sont pas moins, ce qui augmente considérablement la complexité algorithmique des filtres. Modéliser les interactions entre objets soulève la question de la formulation d'hypothèses que l'on effectue sur la scène, ce qui reste encore à formaliser.

Le dernier problème est celui de la dimension de l'espace d'état. Cet enjeu concerne tous les filtres particulaires dont l'état est multi-dimensionnel, et donc particulièrement ceux multi-objets, dans lequel les états des objets sont concaténés pour finalement former le vecteur aléatoire inconnu du système. L'origine de ce problème n'est pas due à l'utilisation d'une méthode de Monte-Carlo pour approcher la loi de filtrage, puisque comme nous l'avons vu dans la section 1.1.1, l'approximation de Monte-Carlo est indépendante de la dimension de l'intégrale, et donc du vecteur d'état. Comme nous le verrons de manière plus détaillée dans le chapitre 5, c'est l'échantillonnage par importance (section 1.1.2.3) qui en est responsable, puisque pour obtenir une qualité d'estimation équivalente, une approximation d'un vecteur de dimension 2M nécessite M^2 particules [Robert et Casella, 2005; MacCormick et Isard, 2000].

1.3.2 Filtres particulaires à modèles multiples

Bien que les méthodes que nous allons exposer ici ne soient pas restreintes au suivi d'objets dans des séquences d'images, nous nous plaçons dans le contexte d'objets dont les dynamiques sont multiples, on dit alors que l'objet effectue des manœuvres. L'ensemble des dynamiques modélisées doit ainsi couvrir l'ensemble des manœuvres que l'objet est capable d'effectuer. L'objectif est alors de prendre en compte ces possibilités de changement de dynamique afin d'être en mesure de suivre l'objet. Les modèles d'interactions multiples ont été proposés dans ce but, et ont été mis en œuvre en les couplant à un filtre de Kalman étendu [Kirubarajan et al., 2001], et un filtre de Kalman sans parfum [Benoudnine et al., 2009]. Le même procédé a également été adapté au filtre particulaire, d'où le nom de filtre particulaire à modèles multiples [McGinnity et Irwin, 2000]. Une extension de ce dernier est l'utilisation d'une variable auxiliaire afin de prendre en compte la dernière observation dans le processus de simulation. Ce procédé est identique à celui du filtre particulaire (section 1.2.6), mais adapté aux modèles multiples [Karlsson et Bergman, 2000]. D'autres techniques effectuant des sauts de Markov dans un système

linéaire [Doucet et al., 2001b; Bergman et al., 2001] et non linéaire [Driessen et Boers, 2005; Andrieu et al., 2003a] ont également été proposées, mais il a été montré dans [Arulampalam et al., 2004] que leurs performances sont moins bonnes que celles des filtres particulaires à modèles multiples.

Ce type de problématique nous intéresse puisque, dans le chapitre 3, nous proposerons une méthode qui semble adaptée aux changements de dynamique, que nous comparerons aux filtres particules à modèles multiples dans le chapitre 4. Nous proposons dans cette section de présenter les filtres particulaires à modèles multiples et à modèles multiples auxiliaires. Une étude comparative de ces deux méthodes, ainsi que des modèles à interactions multiples avec filtre de Kalman étendu et sans parfum, et du modèle à sauts de Markov non linéaire est disponible dans [Arulampalam et al., 2004].

Le principe d'un modèle à sauts est d'ajouter à la variable d'état classique \mathbf{x}_t une variable aléatoire discrète \mathbf{c}_t à valeurs dans $\{1, \ldots, K\}$, formant ainsi ce qu'on appelle un modèle hybride. La variable \mathbf{c}_t désigne l'état, ou régime, dans lequel le système se trouve. La différence entre un filtre particulaire à modèles multiples et un filtre particulaire à sauts de Markov réside dans la manière d'estimer la loi jointe $\mathbb{P}(d\mathbf{x}_t, \mathbf{c}_t | \mathbf{y}_{1:t})$, alors qu'à la base, la modélisation est identique. Ces deux types de méthodes effectuent ce qu'il est courant d'appeler des sauts dans l'espace d'état, en passant ainsi d'une dynamique à une autre. Cela se traduit par le système suivant :

$$\begin{cases} \mathbf{x}_t = f_t(\mathbf{x}_{t-1}, \mathbf{c}_t, \mathbf{v}_t) \triangleq f_t^{\mathbf{c}_t}(\mathbf{x}_{t-1}, \mathbf{v}_t) \\ \mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{w}_t) \end{cases}$$
(1.38)

où $(f_t^i)_{i=1..K}$ et h_t sont des fonctions non linéaires, et où \mathbf{v}_t et \mathbf{w}_t sont deux bruits blancs indépendants. Il existe donc autant de dynamiques qu'il y a de régimes possibles. En principe, et comme nous le verrons bientôt, la probabilité de transition d'une dynamique à une autre est régie par une matrice de transition dont les probabilités sont fixées. Selon l'application, les familles des fonctions dynamiques peuvent être de différentes natures : un modèle à vélocité constante couplé à plusieurs modèles à accélération constante [Mc-Ginnity et Irwin, 2000] ; ou encore un modèle à vélocité constante couplé à des modèles de manœuvre [Kirubarajan et al., 2001; Arulampalam et al., 2004], comme cela sera considéré dans le chapitre 4.

1.3.2.1 Filtre particulaire à modèles multiples

Dans le cadre du filtre particulaire à modèles multiples (MMPF), nous ajoutons à l'estimation de la variable d'état \mathbf{x}_t la variable discrète \mathbf{c}_t à valeurs dans $\{1, \ldots, K\}$. La densité de filtrage s'écrit alors :

$$p(\mathbf{x}_t, \mathbf{c}_t | \mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{c}_t) p(\mathbf{x}_t, \mathbf{c}_t | \mathbf{y}_{1:t-1})}{p(\mathbf{y}_t | \mathbf{y}_{1:t-1})}$$
(1.39)

avec $p(\mathbf{x}_t, \mathbf{c}_t | \mathbf{y}_{1:t-1})$ la densité de prédiction telle que :

$$p(\mathbf{x}_{t}, \mathbf{c}_{t} | \mathbf{y}_{1:t-1}) = \sum_{k=1}^{K} \int_{\mathcal{X}} p(\mathbf{x}_{t-1}, \mathbf{c}_{t-1} = k | \mathbf{y}_{1:t-1}) p(\mathbf{x}_{t} | \mathbf{x}_{t-1}, \mathbf{c}_{t}) \mathbb{P}(\mathbf{c}_{t} | \mathbf{c}_{t-1} = k) d\mathbf{x}_{t-1}$$
(1.40)

menant ainsi à l'approximation particulaire de la loi de filtrage :

$$P_{N}(d\mathbf{x}_{t}, \mathbf{c}_{t}|\mathbf{y}_{1:t}) = \sum_{n=1}^{N} w_{t}^{(n)} \, \delta_{\mathbf{c}_{t}^{(n)}}^{\mathbf{c}_{t}} \, \delta_{\mathbf{x}_{t}^{(n)}}(d\mathbf{x}_{t}) \quad \text{avec} \quad \begin{cases} \mathbf{c}_{t}^{(n)} \sim \mathbb{P}(\mathbf{c}_{t}|\mathbf{c}_{t-1}^{(n)}) \\ \mathbf{x}_{t}^{(n)} \sim p(\mathbf{x}_{t}|\mathbf{x}_{t-1}^{(n)}, \mathbf{c}_{t}^{(n)}) \\ w_{t}^{(n)} = \frac{\tilde{w}(\mathbf{x}_{t}^{(n)}, \mathbf{c}_{t}^{(n)})}{\sum_{m=1}^{N} \tilde{w}(\mathbf{x}_{t}^{(m)}\mathbf{c}_{t}^{(m)})} \end{cases}$$
(1.41)

avec δ_a^b la fonction de Kronecker qui vaut 1 si a = b, 0 sinon, et où $\tilde{w}(\mathbf{x}_t^{(n)}, \mathbf{c}_t^{(n)})$ correspond au poids d'importance séquentiel défini par :

$$\widetilde{w}(\mathbf{x}_{t}^{(n)}, \mathbf{c}_{t}^{(n)}) = w_{t-1}^{(n)} p(\mathbf{y}_{t} | \mathbf{x}_{t}^{(n)}, \mathbf{c}_{t}^{(n)})
= w_{t-1}^{(n)} p(\mathbf{y}_{t} | \mathbf{x}_{t}^{(n)})
= \widetilde{w}(\mathbf{x}_{t}^{(n)})$$
(1.42)

La distribution de transition $\mathbb{P}(\mathbf{c}_t | \mathbf{c}_{t-1})$ est définie par une matrice de transition fixée :

$$\mathbb{P}(\mathbf{c}_t = h | \mathbf{c}_{t-1} = k) \triangleq \alpha_{k,h} \tag{1.43}$$

où la matrice $(\alpha_{k,h})_{(k,h)\in\{1,...,K\}^2}$ définit les probabilités de passer d'un état k à un instant t - 1 à un état h à un instant t. La procédure d'approximation par filtre particulaire à modèles multiples est présentée dans l'algorithme 1.12.

1.3.2.2 Filtre particulaire à modèles multiples auxiliaire

De la même manière que pour le filtre particulaire auxiliaire (section 1.2.6), le filtre particulaire à modèles multiples auxiliaire (AUX-MMPF) a pour objectif d'utiliser la dernière observation dans le processus de simulation, afin d'améliorer l'étape de propagation des particules. Il utilise une variable aléatoire discrète k à valeurs dans $\{1, ..., N\}$ qui indique l'indice de la particule parente. Dans le cadre des modèles multiples, la fonction d'importance $q(\mathbf{x}_t, \mathbf{c}_t, \mathbf{k} | \mathbf{y}_{1:t})$ s'écrit :

$$q(\mathbf{x}_{t}, \mathbf{c}_{t}, \mathbf{k} | \mathbf{y}_{1:t}) = q(\mathbf{x}_{t} | \mathbf{c}_{t}, \mathbf{k}, \mathbf{y}_{1:t}) q(\mathbf{c}_{t}, \mathbf{k} | \mathbf{y}_{1:t})$$

$$\propto p(\mathbf{y}_{t} | \mu_{t}^{(\mathbf{k})}(\mathbf{c}_{t})) p(\mathbf{x}_{t} | \mathbf{x}_{t-1}^{(\mathbf{k})}, \mathbf{c}_{t}) p(\mathbf{c}_{t} | \mathbf{c}_{t-1}^{(\mathbf{k})}) w_{t-1}^{(\mathbf{k})}$$
(1.44)

avec $q(\mathbf{x}_t | \mathbf{c}_t, \mathbf{k}, \mathbf{y}_{1:t}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(\mathbf{k})}, \mathbf{c}_t), q(\mathbf{c}_t, \mathbf{k} | \mathbf{y}_{1:t}) \propto p(\mathbf{y}_t | \mu_t^{(\mathbf{k})}(\mathbf{c}_t)) p(\mathbf{c}_t | \mathbf{c}_{t-1}^{(\mathbf{k})}) w_{t-1}^{(\mathbf{k})}$ et où $\mu_t^{(\mathbf{k})}(\mathbf{c}_t)$ est une caractéristique statistique de \mathbf{x}_t sachant $\mathbf{x}_{t-1}^{(\mathbf{k})}$ et \mathbf{c}_t , telle que son espérance $\mu_t^{(\mathbf{k})}(\mathbf{c}_t) = \mathbb{E}_{p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(\mathbf{k})}, \mathbf{c}_t)}[\mathbf{x}_t]$, ou un échantillon $\mu_t^{(\mathbf{k})}(\mathbf{c}_t) \sim p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(\mathbf{k})}, \mathbf{c}_t)$. Comme dans le filtre particulaire à modèles multiples, la dynamique de la variable aléatoire discrète \mathbf{c}_t est décrite par une matrice de transition fixée $(\alpha_{k,h})_{(k,h)\in\{1,\dots,K\}^2}$.

La simulation selon la fonction d'importance $q(\mathbf{c}_t, \mathbf{k}|\mathbf{y}_{1:t}) \propto p(\mathbf{y}_t|\mu_t^{(\mathbf{k})}(\mathbf{c}_t)) p(\mathbf{c}_t|\mathbf{c}_{t-1}^{(\mathbf{k})}) w_{t-1}^{(\mathbf{k})}$ permet d'obtenir un ensemble de KN couples $\{(\mathbf{c}_t^{(n)}, \mathbf{k}^{(n)})\}_{n=1}^{KN}$, avec K le nombre de régimes. Cela se fait en évaluant, pour chaque particule, la valeur $q(\mathbf{c}_t = i, \mathbf{k}|\mathbf{y}_{1:t})$, avec $i \in \{1, \ldots, K\}$. Un nouveau nuage de N particules est obtenu en rééchantillonnant selon celui décrit par les KN particules $\{(\mathbf{c}_t^{(n)}, \mathbf{k}^{(n)}), w_t^{(n)}\}_{n=1}^{KN}$. Une fois les nouveaux indices et états \mathbf{c}_t obtenus, la simulation

Entrée : Approximation initiale $\mathbb{P}(d\mathbf{x}_0, \mathbf{c}_0) \simeq \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{c}_0^{(n)}}^{\mathbf{c}_0} \delta_{\mathbf{x}_0^{(n)}}(d\mathbf{x}_0)$ **Sortie** : Approximation P_N de la loi *a posteriori* de la trajectoire $\mathbb{P}(d\mathbf{x}_{0:T}, \mathbf{c}_{0:T} | \mathbf{y}_{1:T})$ début pour $t = 1, \ldots, T$ faire • pour $n = 1, \ldots, N$ faire • Générer $\tilde{\mathbf{c}}_t^{(n)} \sim \mathbb{P}(\mathbf{c}_t | \mathbf{c}_{t-1}^{(n)})$ $\circ \text{ Genere } \mathbf{\tilde{x}}_{t}^{(n)} \sim p(\mathbf{x}_{t} | \mathbf{x}_{t-1}^{(n)}, \mathbf{\tilde{c}}_{t}^{(n)}) \\ \circ \text{ Poser } (\mathbf{\tilde{x}}_{0:t}^{(n)}, \mathbf{\tilde{c}}_{0:t}^{(n)}) = \left((\mathbf{x}_{0:t-1}^{(n)}, \mathbf{\tilde{x}}_{t}^{(n)}), (\mathbf{c}_{0:t-1}^{(n)}, \mathbf{\tilde{c}}_{t}^{(n)}) \right)$ Calculer les poids d'importance : $\tilde{w}(\tilde{\mathbf{x}}_{t}^{(n)}) = w_{t-1}^{(n)} p(\mathbf{y}_{t} | \tilde{\mathbf{x}}_{t}^{(n)})$ • pour $n = 1, \ldots, N$ faire • Calculer les poids d'importance normalisés : $w_t^{*(n)} = \frac{\tilde{w}(\tilde{\mathbf{x}}_t^{(n)})}{\sum_{m=1}^N \tilde{w}(\tilde{\mathbf{x}}_t^{(m)})}$ • si $\hat{N}_{eff}(N) < N_{seuil}(N)$ alors • Rééchantillonner $\left\{ \left(\mathbf{x}_{0:t}^{(n)}, \mathbf{c}_{0:t}^{(n)} \right), 1/N \right\}_{n=1}^{N}$ à partir du nuage $\left\{ \left(\tilde{\mathbf{x}}_{0:t}^{(n)}, \tilde{\mathbf{c}}_{0:t}^{(n)} \right), w_t^{*(n)} \right\}_{n=1}^{N}$ selon l'algorithme du rééchantillonnage multinomial 1.9. sinon $\circ \ \operatorname{Poser} \left\{ \left(\mathbf{x}_{0:t}^{(n)}, \mathbf{c}_{0:t}^{(n)} \right), w_t^{(n)} \right\}_{n=1}^N = \left\{ \left(\tilde{\mathbf{x}}_{0:t}^{(n)}, \tilde{\mathbf{c}}_{0:t}^{(n)} \right), w_t^{*(n)} \right\}_{n=1}^N$ • Approximation de la loi marginale au temps t, dite loi de filtrage : $\mathbb{P}(d\mathbf{x}_t, \mathbf{c}_t | \mathbf{y}_{1:t}) \simeq P_N(d\mathbf{x}_t, \mathbf{c}_t | \mathbf{y}_{1:t}) = \sum_{n=1}^N w_t^{(n)} \, \delta_{\mathbf{c}_t^{(n)}}^{\mathbf{c}_t} \, \delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$ • Espérance de Monte-Carlo issue de la loi de filtrage : $\mathbb{E}_{p(\mathbf{x}_t, \mathbf{c}_t | \mathbf{y}_{1:t})}[\phi(\mathbf{x}_t, \mathbf{c}_t)] \simeq \sum_{t=1}^N w_t^{(n)} \phi(\mathbf{x}_t^{(n)}, \mathbf{c}_t^{(n)})$ retourner $P_N(d\mathbf{x}_{0:T}, \mathbf{c}_{0:T} | \mathbf{y}_{1:T}) = \sum_{n=1}^N w_T^{(n)} \, \delta_{\mathbf{c}_{0:T}^{(n)}}^{\mathbf{c}_{0:T}} \delta_{\mathbf{x}_{0:T}^{(n)}}(d\mathbf{x}_{0:T})$

selon $p(\mathbf{x}_t | \mathbf{x}_{t-1}^{(\mathbf{k})}, \mathbf{c}_t)$ se fait directement. Il ne nous reste plus qu'à calculer les nouveaux poids d'importance :

$$\widetilde{w}(\mathbf{x}_{t}^{(n)}, \mathbf{c}_{t}^{(n)}, \mathbf{k}^{(n)}) = w_{t-1}^{\mathbf{k}^{(n)}} \frac{p(\mathbf{y}_{t} | \mathbf{x}_{t}^{(n)}, \mathbf{c}_{t}^{(n)}) p(\mathbf{x}_{t}^{(n)} | \mathbf{x}_{t-1}^{(k)}, \mathbf{c}_{t}^{(n)}) p(\mathbf{c}_{t}^{(n)} | \mathbf{c}_{t}^{\mathbf{k}^{(n)}})}{q(\mathbf{x}_{t}^{(n)}, \mathbf{c}_{t}^{(n)}, \mathbf{k}^{(n)} | \mathbf{y}_{1:t})} \\
= \frac{p(\mathbf{y}_{t} | \widetilde{\mathbf{x}}_{t}^{(n)})}{p(\mathbf{y}_{t} | \mu_{t}^{(\mathbf{k}^{(n)})}(\mathbf{c}_{t}^{(n)}))}$$
(1.45)

A l'issue de l'étape de mise à jour des poids, les indices $\mathbf{k}^{(n)}$ sont ignorés. Le filtre particulaire à modèles multiples auxiliaire est présenté dans l'algorithme 1.13, dans lequel nous avons omis l'approximation de la loi de filtrage ainsi que l'estimation de l'espérance de Monte-Carlo, qui restent identiques à celles formulées dans l'algorithme 1.12, par manque de place.

Ce filtre combine l'avantage du MMPF, qui décrit précisément la dynamique complète d'un objet effectuant des manœuvres, et l'AUX-PF, qui utilise la dernière observation pour propager les particules. Une conséquence directe est que le coût de l'algorithme du AUX-MMPF est en O(N(K + 1)), ce qui peut être élevé, dans la mesure où en traitement d'images, l'évaluation de la vraisemblance constitue généralement l'étape la plus coûteuse en termes de temps de calcul. De plus, le rééchantillonnage effectué à l'issue de la première étape de l'algorithme considère non pas N mais KN particules, ce qui représente également un surcoût de calcul à ne pas négliger (la procédure de rééchantillonnage pour un échantillon étant linéaire par rapport au nombre de particules).

1.4 Conclusion

Nous avons, dans ce chapitre, fait un rapide panorama des méthodes d'approximation de loi par Monte-Carlo. Nous les avons décomposées en deux catégories : les méthodes de Monte-Carlo classiques, qui comprennent les méthodes non itératives de simulations de lois, qui sont la transformée inverse, l'algorithme d'acceptation / rejet et l'échantillonnage par importance, et les méthodes de Monte-Carlo par chaînes de Markov, telles que l'algorithme de Metropolis-Hastings, et un cas particulier, celui de l'échantillonneur de Gibbs. La définition de tous ces concepts permettent de mieux appréhender les enjeux qui interviennent dans la procédure d'approximation d'une loi.

La seconde catégorie étudiée est celle des méthodes de Monte-Carlo séquentielles, communément appelées filtres particulaires. Une littérature particulièrement féconde a été proposée à ce sujet depuis la deuxième moitié des années 1990. Nous avons soulevé un certain nombre de difficultés liées à la mise en œuvre d'un filtre particulaire pour le suivi d'objets dans des séquences d'images. Parmi elles, nous avons choisi de nous confronter, dans le cadre de cette thèse, à quatre problèmes classiques, en essayant de répondre à des questions ouvertes.

Le premier est celui de la modélisation de la dynamique qui est souvent inconnue en pratique. Ainsi, une solution quasi systématique consiste à considérer une dynamique simple, de manière à ne pas contraindre le suivi, et à surestimer les paramètres de bruit, dans le but de ne pas perdre la trace de l'objet. Plutôt que de faire cela, comme première contribution originale, nous proposons d'intégrer au suivi des informations structurelles, qui décrivent les relations d'un objet entre deux temps successifs. Cette information est

Algorithme 1.13: Filtre particulaire à modèles multiples avec l'utilisation d'une variable auxiliaire (AUX-MMPF).

Entrée : Approximation initiale $\mathbb{P}(d\mathbf{x}_0, \mathbf{c}_0) \simeq \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{c}_0^{(n)}}^{\mathbf{c}_0} \delta_{\mathbf{x}_0^{(n)}}(d\mathbf{x}_0)$ **Sortie** : Approximation P_N de la loi *a posteriori* de la trajectoire $\mathbb{P}(d\mathbf{x}_{0:T}, \mathbf{c}_{0:T} | \mathbf{y}_{1:T})$ **début**

```
pour t = 1, \ldots, T faire
          1. Première étape :
                    • pour n = 1, \ldots, N faire
                            • pour i = 1, \ldots, K faire
                                    \diamond Calculer \mu_t^n(i)

    Calculer les poids d'importance :

                                                                                               \tilde{w}(\mu_t^{(n)}(i)) = w_{t-1}^{(n)} p(\mathbf{y}_t | \mu_t^{(n)}(i))
                   • pour n = 1, \ldots, N faire
                            • pour i = 1, \ldots, K faire

    Calculer les poids d'importance normalisés :

                                                                                                 w_t^{*(n(i))} = \frac{\tilde{w}(\mu_t^{(n)}(i))}{\sum_{m=1}^N \tilde{w}(\mu_t^{(m)}(i))}
                    • pour n = 1, \ldots, N faire
                          \circ \text{ Générer le couple } (\mathbf{c}_t^{(n)}, \mathbf{k}^{(n)}) \sim \sum_{i=1}^N \sum_{i=1}^K \delta_i^{\mathbf{c}_t} \, \delta_m^{\mathbf{k}} \, w_t^{*(m(i))}
          2. Etape SIR:
                   • pour n = 1, \dots, N faire
                          \circ \text{ Générer } \mathbf{x}_{t}^{(n)} \sim p(\mathbf{x}_{t} | \mathbf{x}_{t-1}^{(k^{(n)})}, \mathbf{c}_{t}^{(n)}) \\ \circ \text{ Poser } (\mathbf{x}_{0:t}^{(n)}, \mathbf{c}_{0:t}^{(n)}) = \left( (\mathbf{x}_{0:t-1}^{(k^{(n)})}, \mathbf{x}_{t}^{(n)}), (\mathbf{c}_{0:t-1}^{(k^{(n)})}, \mathbf{c}_{t}^{(n)}) \right)
                            • Calculer les poids d'importance :
                                                                          \tilde{w}(\mathbf{x}_{t}^{(n)}, \mathbf{c}_{t}^{(n)}, \mathbf{k}^{(n)}) = \frac{p(\mathbf{y}_{t} | \mathbf{x}_{t}^{(n)})}{p(\mathbf{y}_{t} | \mu_{\star}^{(\mathbf{k}^{(n)})}(\mathbf{c}_{\star}^{(n)}))}
                   • pour n = 1, \ldots, N faire
                            • Calculer les poids d'importance normalisés :
                                                                                 w_t^{(n)} = \frac{\tilde{w}(\mathbf{x}_t^{(n)}, \mathbf{c}_t^{(n)}, \mathbf{k}^{(n)})}{\sum_{m=1}^N \tilde{w}(\mathbf{x}_t^{(m)}, \mathbf{c}_t^{(m)}, \mathbf{k}^{(m)})}
retourner P_N(d\mathbf{x}_{0:T}, \mathbf{c}_{0:T}|\mathbf{y}_{1:T}) = \sum_{n=1}^N w_T^{(n)} \, \delta_{\mathbf{c}_{0:T}^{(n)}}^{\mathbf{c}_{0:T}} \, \delta_{\mathbf{x}_{0:T}^{(n)}}^{\mathbf{c}_{0:T}} (d\mathbf{x}_{0:T})
```

vague, est donc modélisée dans un formalisme flou, comme nous le décrirons dans le chapitre 2. Elle permet de décrire la dynamique de l'objet de manière imprécise, ce qui possède l'avantage de ne pas la contraindre et ainsi être robuste aux changements brusques de la trajectoire, tout en exploitant la structure des informations spatiales floues considérées, qui permettent de définir une dynamique dans le but de propager de manière efficace les particules d'un instant à l'autre. Le dernier avantage de cette modélisation souple est la possibilité de l'adapter à des informations spatiales générales, comme nous le proposerons dans le cadre des formes floues.

Le chapitre 3 traite de l'intégration d'informations spatiales floues dans le filtre particulaire. Nous y proposerons également une nouvelle fonction d'importance, qui dépend de la trajectoire des états, de manière à lever les ambiguïtés spatio-temporelles locales, ainsi que de la dernière observation, afin de mieux propager le nuage de particules. L'utilisation de la trajectoire pour générer les particules, n'avait, à notre connaissance, jamais été faite auparavant. Cette fonction d'importance est totalement dédiée au formalisme flou que nous proposons, et bien adaptée dans le cas où la dynamique de l'objet est inconnue, ou au moins mal connue. Dans ce cadre, nous aurons l'occasion dans le chapitre 4 de comparer les résultats obtenus avec d'autres filtres classiques de suivi (AUX-PF, MMPF, AUX-MMPF), dont certains utilisent également l'observation dans l'étape de propagation des particules (AUX-PF, AUX-MMPF).

La troisième contribution de cette thèse concerne le filtrage particulaire multi-objets. Comme nous l'avons vu, celui-ci souffre d'un problème lié à la dimension de l'espace d'état. Comme nous le verrons dans le chapitre 5, une méthode pour le résoudre tente d'éviter le surcoût du nombre de particules nécessaires à l'approximation de loi jointe de filtrage. Nous proposerons une amélioration de cette méthode appelée échantillonnage partitionné et proposée dans [MacCormick et Blake, 2000; MacCormick et Isard, 2000], qui rend le suivi plus robuste, particulièrement dans les cas d'occultations ou plus généralement lorsque l'estimation d'un ou de plusieurs objets est incertaine. Nous proposerons également d'intégrer au filtre particulaire multi-objets des contraintes spatiales floues entre les objets. L'ajout de ces informations structurelles permet de contraindre la configuration globale des objets, et ainsi de traduire facilement les hypothèses formulées sur la séquence à suivre.

Enfin, le dernier problème auquel nous avons choisi de nous intéresser est celui de l'utilisation de plusieurs modalités. Dans ce domaine, nous avons constaté l'émergence de deux problématiques courantes : celle de l'utilisation de plusieurs vraisemblances, issues d'un même capteur, et celle de l'utilisation de plusieurs capteurs. Les travaux correspondant traitent de la fusion d'informations, au sens probabiliste, souvent selon des critères de pertinence de la modalité considérée, qui peuvent être mal estimés. Nous proposerons un travail préliminaire dans le chapitre 6, qui consiste non pas à considérer plusieurs caractéristiques ni même plusieurs capteurs, mais à considérer des vraisemblances à modèles multiples. Nous partons du principe que l'état d'un objet influence le modèle de vraisemblance à considérer. Par exemple, dans le cadre de suivi d'une bouche qui s'ouvre et se referme, nous pourrons définir par une loi de mélange deux vraisemblances correspondant à ces deux états, dont les paramètres sont liés à l'état de la particule, bouche ouverte ou fermée. Ce type d'information n'a, à notre connaissance, pas été exploité jusqu'ici.

Chapitre 2

Modélisation d'informations spatiales floues

Sommaire

2.1	Inform	rmations spatiales floues 5										
	2.1.1	Information floue										
	2.1.2	Fonctions floues										
	2.1.3	Stratégies de fusion										
	2.1.4	Ensemble flou des événements possibles										
2.2	Relations spatiales floues											
	2.2.1	Mise en œuvre										
		2.2.1.1 Cas monodimensionnel										
		2.2.1.2 Cas multidimensionnel										
	2.2.2	Transition										
		2.2.2.1 Cas monodimensionnel										
		2.2.2.2 Cas multidimensionnel										
2.3	Informations de formes floues											
	Mise en œuvre générique 66											
	2.3.2	Application à la modélisation de la bouche 68										
		2.3.2.1 Contraintes de formes										
		2.3.2.2 Transition										
2.4	Concl	usion										

Les informations spatiales floues permettent de représenter une configuration spatiale entre une ou plusieurs entités. Dans ce chapitre, nous considérerons les informations spatiales définies entre un seul objet à un instant donné et ce même objet à un instant passé. On cherche à modéliser des informations spatiales floues, pour définir des relations spatio-temporelles floues, ou encore des informations de formes floues. Le but est d'intégrer au suivi des informations de haut niveau, de sémantiques fortes, définies de manière souple et dans un cadre unifié, afin de facilement adapter le modèle à différents types d'applications (mouvement, forme, ...).

Après avoir défini les informations spatiales floues dans un contexte général (section 2.1, nous étudierons dans la section 2.2 la formalisation des relations spatiales floues puis celle d'informations spatiales, telles que les formes floues (section 2.3).

2.1 Informations spatiales floues

La théorie des ensembles flous permet de modéliser des connaissances qualitatives, vagues ou imprécises, souvent exprimées sous forme linguistique [Dubois et Prade, 1980]. Par exemple, lorsque nous réalisons un sondage auprès d'un ensemble de personnes, en leur demandant quelle est leur taille de manière qualitative, nous pourrions aisément regrouper les réponses obtenues selon une granulométrie bien choisie, telle que : *petit, moyen* et *grand*. Dans la mesure où ces valeurs linguistiques ne sont que subjectives, donc dépendent des origines, du genre, *etc.* de chacun, il est très probable que les classes obtenues, c'est-à-dire les ensembles flous, comportent des valeurs communes. Là où une analyse fréquentiste indiquerait avec quelle probabilité un individu tiré aléatoirement dans la population peut mesurer une certaine taille, l'approche floue apporte, généralement, une valeur possibiliste.

En traitement d'images et en vision, et plus particulièrement en suivi d'objets, il est possible de modéliser certaines caractéristiques, ou certains comportements, par une représentation floue. Ainsi, il est par exemple possible de décrire le volume d'un objet par des ensembles flous, avec une granulométrie fixée qui influe sur la précision de la représentation. De même la distance entre les positions d'un objet à deux instants différents peut être caractérisée par les valeurs *proche*, à *moyenne distance*, *éloigné*, dont la sémantique est définie par des ensembles flous sur \mathbb{R}^+ . Ici encore, l'information à modéliser est déterministe mais vague, rendant les ensembles flous appropriés puisqu'ils permettent très naturellement d'exprimer ce type d'imperfection. D'une manière similaire, il est possible d'introduire des informations de formes floues, chaque ensemble flou représentant une posture (figure 2.1). Par exemple, la posture *debout* d'un individu pourra être définie par un ensemble flou, dont les éléments d'appartenance non nulle, c'est-à-dire les formes, seront des représentations possibles de cette posture. Dans ce chapitre, nous proposons une structure générale pour modéliser ce type d'information.



FIG. 2.1 – De l'utilisation d'informations spatiales floues en suivi. Des informations spatiales floues passées sont utilisées pour estimer l'état courant. Elles peuvent être de différentes natures : de forme, de taille, d'orientation, *etc.*

Un concept flou est défini par une famille d'ensembles flous. Chacun de ces ensembles correspond à une valeur possible du concept flou, et contient des éléments qui sont des représentations possibles de cette valeur. Par exemple, $\pi/4$ est une représentation possible de la valeur *nord-est* du concept flou d'orientation, selon une valeur d'appartenance définie, qui indique selon quel degré l'élément traduit la valeur considérée.

Considérons une famille de K ensembles flous $\{\mathcal{A}_k\}_{k=1}^K$ définis par les fonctions d'appartenance $\{\mu_k\}_{k=1}^K$ telles que $\forall k \in \{1, \ldots, K\}, \mu_k : \mathcal{O} \to [0, 1], \mathcal{O} \subset \mathbb{R}^{n_0}$, avec \mathcal{O} l'espace des éléments flous et n_0 sa dimension. Nous associons à chaque élément $\mathbf{o} \in \mathcal{O}$ un degré d'appartenance $\mu_k(\mathbf{o})$ à l'ensemble flou \mathcal{A}_k . Par exemple, si l'ensemble \mathcal{A}_k définit les éléments impossibles d'un certain concept flou, alors $\mu_k(\mathbf{o})$ indique avec quel degré \mathbf{o} traduit cette impossibilité. Par exemple, l'élément $-\pi/4$ traduit avec un degré fort l'impossibilité de la valeur *sud* du concept flou orientation, dans le cas où nous souhaiterions contraindre celle-ci.

Dans la terminologie floue, la famille de fonctions $\{\mu_k\}_{k=1}^K$ correspond à une variable linguistique dans laquelle chaque fonction μ_k donne la sémantique d'une valeur linguistique possible de la variable. La granularité est définie en fonction de l'application. Ainsi, en considérant une relation spatiale d'orientation floue, les valeurs linguistiques pourraient être $A_1 = Est$, $A_2 = Nord$, $A_3 = Ouest$, $A_4 = Sud$, définies par la famille de fonctions d'appartenance $\{\mu_k\}_{k=1}^4$.

2.1.2 Fonctions floues

Nous définissons dans cette section une manière générale de modéliser les fonctions d'appartenance associées à un certain concept flou. Leur définition fait intervenir certains concepts génériques qui permettent par exemple de représenter des relations spatiales floues et des formes floues. Des exemples et des illustrations de ces concepts flous seront donnés dans les sections suivantes, qui traitent de la modélisation des relations spatiales et des formes floues.

Les fonctions floues $\{\mu_k\}_{k=1}^K$ peuvent être fixées, soit par une connaissance *a priori*, donc liée à l'application, soit de manière générique, par apprentissage. C'est dans ce dernier contexte que nous nous plaçons maintenant. Considérons M^k éléments, que nous appelons observations, appartenant à la classe k. Soit $\{\mathbf{o}_m^k\}_{m=1}^M$ ces observations, telles que $\forall m \in \{1, \ldots, M^k\}, \mathbf{o}_m^k \in \mathcal{O}$. Afin de construire les fonctions $\{\mu_k\}_{k=1}^K$, il est tout d'abord nécessaire de définir la famille $\{\tilde{\mu}_k\}_{k=1}^K$, dont chacune des fonctions est indépendante des autres. La fonction $\tilde{\mu}_k$ correspond donc à μ_k sans la connaissance des autres fonctions d'appartenance $\{\tilde{\mu}_h\}_{\forall h,h\neq k}$. Elle s'écrit simplement comme une combinaison floue des observations $\{\mathbf{o}_m^k\}_{m=1}^{M^k}$:

$$\tilde{\mu}_k(\mathbf{o}) = \Phi^k(g_{\mathbf{o}_1^k}(\mathbf{o}), \dots, g_{\mathbf{o}_{M^k}^k}(\mathbf{o}))$$
(2.1)

avec $\Phi^k : [0,1]^{M^k} \to [0,1]$ un opérateur de fusion floue, que nous détaillerons dans la section 2.1.3, et $g_{\mathbf{o}_m^k} : \mathcal{O} \to [0,1]$ une fonction associée à \mathbf{o}_m^k :

$$g_{\mathbf{o}_{m}^{k}}(\mathbf{o}) = \top \left[P_{\mathbf{o}_{m}^{k}}(\mathbf{o}), C_{\mathbf{o}_{m}^{k}}(\mathbf{o}) \right]$$
(2.2)

avec \top une t-norme (conjonction floue) [Dubois et Prade, 1980], $P_{\mathbf{o}_m^k} : \mathcal{O} \to [0, 1]$ un patron flou générique connu et fixé et $C_{\mathbf{o}_m^k} : \mathcal{O} \to [0, 1]$ des contraintes sur la forme de la

fonction floue finale. Ainsi la fonction $g_{\mathbf{o}_m^k}$ représente l'équivalent flou de l'observation \mathbf{o}_m^k , sous un patron générique $P_{\mathbf{o}_m^k}$ contraint par $C_{\mathbf{o}_m^k}$, qui garantit la cohérence globale de $g_{\mathbf{o}_m^k}$, par rapport à \mathbf{o}_m^k . Nous appelons patron générique un ensemble flou non contraint, selon lequel il est aisé de générer un échantillon, et qui n'est pas spécifique à une application. Le principe est illustré sur la figure 2.2.



FIG. 2.2 – Principe du raisonnement par contraintes floues. Une boule floue P_m^k est d'abord considérée, puis une contrainte (ici une localisation spatiale) C_m^k et, à l'aide d'une t-norme, le résultat contraint g_m^k . Les ensembles flous sont représentés dans le domaine de l'image. Le niveau de gris associé à chaque pixel traduit l'appartenance de l'élément qu'il représente à l'ensemble flou, un niveau de gris sombre indiquant une appartenance forte.

Ces fonctions peuvent être connues *a priori*, ou apprises, comme nous le verrons dans la section 2.3. En l'absence de contrainte, $C_{\mathbf{o}_m^k}$ correspond à l'élément neutre de la t-norme, c'est-à-dire à la valeur 1. Le patron flou $P_{\mathbf{o}_m^k}$ peut par exemple correspondre, dans le cas où $\mathcal{O} \subset \mathbb{R}^L$, à une boule floue de dimension L, qui est une généralisation possible du patron 1D trapézoïdal, avec l'appartenance qui est fonction de la distance euclidienne $(\|.\|_2)$ entre le centre \mathbf{o}_m^k et l'élément considéré :

$$P_{\mathbf{o}_{m}^{k}}(\mathbf{o}) = \begin{cases} 1 & \text{si } \|\mathbf{o} - \mathbf{o}_{m}^{k}\|_{2} \in [0, b] \\ 1 - \frac{\|\mathbf{o} - \mathbf{o}_{m}^{k}\|_{2} - b}{a} & \text{si } \|\mathbf{o} - \mathbf{o}_{m}^{k}\|_{2} \in]b; b + a[0 & \text{sinon} \end{cases}$$

où 2(a + b) et 2*b* sont les longueurs du support et du noyau (l'ensemble des éléments dont le degré d'appartenance à l'ensemble considéré vaut 1), respectivement. Dans cet exemple, $P_{\mathbf{o}_m^k}(\mathbf{o})$ correspond bien à une version floue de l'observation \mathbf{o}_m^k , qui a une appartenance maximale autour de cette valeur, et qui décroit ensuite, dès un certain seuil passé, de manière proportionnelle à la distance au centre. Enfin, nous définissons la fonction floue μ_k finale comme une combinaison des fonctions $\tilde{\mu}_k$ et $\{\tilde{\mu}\}_{\setminus k} \triangleq$ $\{\tilde{\mu}_h\}_{h \in \{1,...,K\} \setminus \{k\}}$, par un opérateur de fusion $\Psi^k : [0,1]^K \to [0,1] :$

$$\mu_k(\mathbf{o}) = \Psi^k\left(\tilde{\mu}_k(\mathbf{o}), \{\tilde{\mu}_h(\mathbf{o})\}_{\setminus k}\right)$$
(2.3)

L'opérateur de fusion Ψ^k intervient typiquement lorsque des éléments ont des appartenances non nulles à plusieurs classes ; c'est par exemple le cas pour $\pi/8$ avec les valeurs *est et nord-est* du concept d'orientation. La fonction μ_k intègre à la fois une information interne de la classe k, avec l'opérateur de fusion intra-classes Φ^k , et une information partagée par toutes les classes, représentée par l'opérateur inter-classes Ψ^k .

Il devient alors possible de définir des interactions entre les ensembles flous en calculant leur degré d'intersection, contraint par un niveau minimum. Soit R(k,h) le degré d'intersection contraint entre les ensembles μ_k et μ_h :

$$R(k,h) = \bot \left[\Xi^{kh}(\mu_k,\mu_h),\epsilon^{kh}\right]$$
(2.4)

avec \perp une t-conorme (disjonction floue) [Dubois et Prade, 1980], $\Xi^{kh}(\mu_k, \mu_h)$ le degré d'intersection des ensembles flous μ_k et μ_h , et ϵ^{kh} le seuil minimal d'intersection des deux ensembles, connu et fixé. Il existe de nombreux opérateurs calculant le degré d'intersection de deux ensembles flous, voir par exemple [Bloch, 2005] pour un descriptif des opérateurs existants, ou les sections 2.2 et 2.3 pour ceux que nous utiliserons dans nos expériences.

2.1.3 Stratégies de fusion

Comme nous venons de le voir, les ensembles flous $\{\mu_k\}_{k=1}^K$ sont obtenus en combinant différentes stratégies de fusion via les opérateurs $\{\Phi^k\}_{k=1}^K$, $\{\Psi^k\}_{k=1}^K$. Le premier est appelé opérateur de fusion intra-classes, car il se réfère à la fusion d'observations issues d'une même classe, tandis que le second fusionne des données de différentes classes, et correspond donc à un opérateur de fusion inter-classes. Il est ainsi possible de décliner une liste non exhaustive des différentes stratégies de fusion envisageables (le lecteur peut se référer à [Dubois et Prade, 1980; Yager, 1980, 1991; Bloch, 1996] pour une description plus complète de ces opérateurs de fusion) :

Stratégies intra-classes : Φ

- Fusion disjonctive en utilisant des t-conormes : chaque observation représente une vision partielle de l'ensemble des possibles, et donc leur réunion couvre du mieux possible cet ensemble. Le degré μ_k(o) représente alors dans quelle mesure o ressemble à au moins une des observations. C'est ce type de fusion que nous privilégierons dans nos expériences.
- Fusion conjonctive en utilisant des t-normes, intéressante dans au moins deux situations. Lorsque nous avons des observations multiples d'un même phénomène, chacune entâchée d'imprécision, nous pouvons avoir intérêt à dilater les représentations floues correspondantes, pour qu'elles englobent bien la vérité et pour éviter les conflits entre elles. Dans ce cas la conjonction ne retient que les parties communes, données par toutes les observations, et permet alors de réduire l'imprécision. Autre situation, lorsque les observations ne donnent pas les possibles mais au contraire des contraintes (par exemple les valeurs impossibles ou interdites) : dans ces cas nous cherchons une conjonction des complémentaires des contraintes. Avec une fusion conjonctive, μ_k(o) mesure alors l'adéquation entre o et l'ensemble des observations.
- Fusion de type compromis en utilisant une somme ou une moyenne : cela permet une compensation entre les observations, considérées par exemple comme des évaluations d'une situation, chaque mesure pouvant être imparfaite, nous pouvons les moyenner pour compenser les imperfections de chacune. Avec un opérateur additif, nous obtenons une approche fréquentiste, et ainsi rejoignons les questions d'estimation statistique.

Les stratégies intra-classes sont illustrées sur la figure 2.3.

Stratégies inter-classes : Ψ

• Fusion exclusive : si un élément a une appartenance partagée entre plusieurs classes, nous considérons que c'est une impossibilité sémantique. Cela peut être appliquable à la modélisation d'états binaires, pour lesquels il n'existe pas de valeurs intermédiaires. Dans ce cas, une fusion possible consiste à conserver l'état



FIG. 2.3 – Stratégies de fusion intra-classes avec (a) deux ensembles flous décrivant la même classe avant fusion, (b) fusion disjonctive (t-conorme = \max), (c) fusion conjonctive (t-norme = \min) et (d) fusion de type compromis (moyenne).

dont la valeur d'appartenance est la plus forte, ou celle qui est la plus proche d'une observation en cas d'égalité.

- Fusion normalisatrice : nous normalisons l'appartenance de chaque élément à une classe par la somme des appartenances à toutes les classes.
- Aucune : c'est le comportement par défaut, chaque classe est considérée indépendamment des autres.

Les stratégies inter-classes sont illustrées sur la figure 2.4. Ainsi, en fonction de certaines propriétés désirées, ou selon une sémantique particulière des classes, il est possible de combiner ces différentes stratégies afin d'arriver à la représentation souhaitée.



FIG. 2.4 – Stratégies de fusion inter-classes avec (a) deux ensembles flous décrivant deux classes différentes avant fusion, (b) fusion exclusive, (c) fusion normalisatrice et (d) aucune fusion.

2.1.4 Ensemble flou des événements possibles

Avant d'étudier l'utilisation en pratique d'informations spatiales floues, nous avons besoin d'introduire le concept d'ensemble flou Z des événements possibles, dont la fonction d'appartenance est notée μ_{Z} . Typiquement, celui-ci dépend de la collection considérée $\{\mathcal{A}_k\}_{k=1}^K$, et joue un rôle important puisqu'il conditionne la manière dont les informations spatiales floues seront intégrées dans un contexte probabiliste (section 3.2). Nous présentons ici une liste non exhaustive de stratégies pour définir Z:

- Stratégie disjonctive :
 - Union : $\mathcal{Z} = \bigcup_{k=1}^{K} \mathcal{A}_k$

$$\Leftrightarrow \mu_{\mathcal{Z}}(\mathbf{o}) = \underset{k=1}{\overset{K}{\perp}} \mu_k(\mathbf{o}) \; \forall \mathbf{o} \in \mathcal{C}$$

– Somme algébrique (cas particulier de l'union) : $\mathcal{Z} = \mathcal{A}_1 \oplus \cdots \oplus \mathcal{A}_K$

$$\Leftrightarrow \mu_{\mathcal{Z}}(\mathbf{o}) = \sum_{k=1}^{K} \mu_k(\mathbf{o}) - \prod_{k=1}^{K} \mu_k(\mathbf{o}) \ \forall \mathbf{o} \in \mathcal{O}$$

- Stratégie conjonctive : Intersection : $\mathcal{Z} = \bigcap_{k=1}^{K} \mathcal{A}_k$

$$\Leftrightarrow \mu_{\mathcal{Z}}(\mathbf{o}) = \mathop{\top}\limits_{k=1}^{K} \mu_k(\mathbf{o}) \; \forall \mathbf{o} \in \mathcal{O}$$

– Produit (cas particulier de l'intersection) : $\mathcal{Z} = \prod_{k=1}^{K} \mathcal{A}_k$

$$\Leftrightarrow \mu_{\mathcal{Z}}(\mathbf{o}) = \prod_{k=1}^{K} \mu_k(\mathbf{o}) \; \forall \mathbf{o} \in \mathcal{O}$$

- Stratégie de type compromis :
 - Moyenne arithmétique : $\mathcal{Z} = \frac{1}{K} \sum_{k=1}^{K} \mathcal{A}_k$

$$\Leftrightarrow \mu_{\mathcal{Z}}(\mathbf{o}) = \frac{1}{K} \sum_{k=1}^{K} \mu_k(\mathbf{o}) \; \forall \mathbf{o} \in \mathcal{O}$$

- Additive (dans le cas où $\forall \mathbf{o} \in \mathcal{O}, \sum_{k=1}^{K} \mu_k(\mathbf{o}) \leq 1$) : $\mathcal{Z} = \sum_{k=1}^{K} \mathcal{A}_k$

$$\Leftrightarrow \mu_{\mathcal{Z}}(\mathbf{o}) = \sum_{k=1}^{K} \mu_k(\mathbf{o}) \; \forall \mathbf{o} \in \mathcal{O}$$

avec \top une t-norme et \perp une t-conorme.

Nous allons maintenant appliquer le modèle général présenté à la modélisation de relations spatiales floues entre les positions d'un objet à des instants différents, donc donnant une information structurelle, et à l'information de formes floues caractérisant les objets eux-mêmes, qui font l'objet des deux prochaines sections.

2.2 **Relations spatiales floues**

Un des enjeux classiques en suivi est de modéliser, d'une manière aussi proche que possible de la réalité, la dynamique d'un objet, afin d'être capable de prédire, à partir d'un certain temps t, son état à un instant $t + \Delta$.

Chaque relation spatiale est considérée comme une variable linguistique, prenant un petit nombre de valeurs linguistiques [Zadeh, 1975]. La granulométrie de cette représentation peut être définie par l'application. La sémantique de chaque valeur linguistique est définie par un ensemble flou sur le domaine de la variable (par exemple $[0, 2\pi]$ pour l'orientation, \mathbb{R}^+ pour les distances, \mathbb{R}^2 pour la vélocité 2D, *etc.*).

Par exemple, pour l'orientation, nous pourrions considérer huit valeurs linguistiques : est, nord-est, nord, nord-ouest, ouest, sud-ouest, sud, sud-est (figure 2.6). La sémantique de chaque valeur est définie par une fonction d'appartenance trapézoïdale sur l'axe des angles. L'ensemble des fonctions d'appartenance forme une partition floue de $[0, 2\pi]$. La possibilité de superposition entre les fonctions d'appartenance est un paramètre de notre méthode. Ici nous supposons que seulement deux "relations successives" peuvent se superposer (par exemple, nord et nord-est).

Pour les distances, trois valeurs pourraient par exemple être considérées : proche, à moyenne distance, loin ; et neuf pour la vélocité (déplacement) : de en haut à gauche à en bas *à droite*. Par exemple, si nous souhaitons représenter la dynamique d'un objet grâce à un système de coordonnées polaires, nous modéliserons le couple orientation / distance. Ces relations peuvent également être représentées dans le domaine spatial [Bloch, 2005], ce qu'illustre la figure 2.5, en utilisant des formes de fonctions floues différentes. Notons que la représentation des ensembles flous dans le domaine spatial (de l'image) n'est donnée ici qu'à titre illustratif, puisqu'en pratique, nous ne les manipulons que dans le domaine des variables ($[0, 2\pi], \mathbb{R}^+, ...$).



FIG. 2.5 – Relations spatiales floues représentées dans le domaine des variables (a-c) et de l'image (d-f). Une valeur sombre représente une appartenance maximale ; (a,d) valeur *nord* du concept flou d'orientation en utilisant un patron trapézoïdal, (b,e) valeur *nord-est* du concept flou d'orientation en utilisant un patron triangulaire et (c,f) valeur *à moyenne distance* du concept flou de distance en utilisant un patron binaire.

2.2.1 Mise en œuvre

Les relations spatiales forment une partition floue de l'espace considéré. Elles peuvent être construites en considérant une stratégie de fusion intra-classes disjonctive (l'ensemble disjoint des éléments définit la valeur linguistique) et une stratégie inter-classes normalisatrice (afin de garantir des propriétés de normalisation). Néanmoins, elles sont construites dans cette section directement, sans utiliser d'observations (qui est un concept général), tout en étant cohérentes avec les propriétés mentionnées.

2.2.1.1 Cas monodimensionnel

Afin de définir les relations spatiales monodimensionnelles, reprenons tout d'abord la définition de partition uniforme floue forte donnée notamment dans [Perfilieva, 2006; Loquin et Strauss, 2008].

Définition 2.1 Soit $\mathcal{O} = [I_1, I_2] \subset \mathbb{R}$ le domaine de la partition. Soit $m_1 < m_2 < \cdots < m_K$ les K points fixés de \mathcal{O} que l'on appelle nœuds, tels que $m_1 = I_1$ et $m_K = I_2$, avec $K \ge 2$, et $\forall k \neq K, m_{k+1} - m_k = h$ = constante. La famille des K ensembles flous $\{\mathcal{A}_k\}_{k=1}^K$ définis par leurs fonctions d'appartenance $\{\mu_k\}_{k=1}^K$ sur \mathcal{O} forme une partition uniforme floue forte si elle satisfait les propriétés suivantes :

1.
$$\forall k \in \{1, \dots, K\}, \mu_k(m_k) = 1,$$

2. $\forall k \in \{1, \dots, K\}, si \ x \notin [m_{k-1}, m_{k+1}], \mu_k(x) = 0,$

- 3. $\forall k \in \{1, \dots, K\}, \mu_k(x) \text{ a une croissance monotone sur } [m_{k-1}, m_k] \text{ et une décroissance monotone sur } [m_k, m_{k+1}],$
- 4. $\forall x \in \mathcal{O}, \sum_{k=1}^{K} \mu_k(x) = 1,$
- 5. $\forall k \in \{2, \dots, K-1\}, \forall x \in [0, h], \mu_k(m_k x) = \mu_k(m_k + x),$
- 6. $\forall k \in \{2, \dots, K-1\}, \forall x \in [m_k, m_{k+1}], \mu_k(x) = \mu_{k-1}(x-h),$
- 7. $\forall k \in \{2, \dots, K-1\}, \forall x \in [m_{k-1}, m_k], \mu_k(x) = \mu_{k+1}(x+h).$

Un exemple de partition uniforme floue forte est illustré sur la figure 2.6.



FIG. 2.6 – Exemple de partition uniforme floue forte, en utilisant des fonctions d'appartenances de formes trapézoïdales et semi-trapézoïdales en considérant K = 4 ensembles flous $\{A_k\}_{k=1}^4$ de nœuds $\{m_k\}_{k=1}^4$.

Nous utilisons maintenant la définition de partition uniforme floue forte afin de définir les relations spatiales. Celles-ci doivent satisfaire les propriétés 1 et 4 à des fins de normalisation et la propriété 2 afin d'assurer que seuls deux ensembles voisins auront un degré d'intersection non nul. Bien que non nécessaires, les propriétés 3, 5, 6 et 7 permettent de faciliter leur manipulation, par exemple en considérant des ensembles flous dont les fonctions d'appartenance correspondent à des trapèzes de formes identiques. Afin d'introduire les relations spatiales floues, nous ajoutons une autre propriété, qui contraint les intégrales sur O des fonctions d'appartenance à être identiques :

8. $\forall (h,k) \in \{1,\ldots,K\}^2$,

$$\int_{\mathcal{O}} \mu_h(x) \, dx = \int_{\mathcal{O}} \mu_k(x) \, dx \tag{2.5}$$

Cette propriété sera nécessaire lors de l'intégration des relations spatiales dans le filtre particulaire, afin de garantir une exploration de l'espace O uniforme (voir la section 3.1 pour plus de détails). Nous différencions maintenant les relations spatiales de domaine de définition de la forme $[a + 2i\pi, a + 2(i+1)\pi], a \in \mathbb{R}, i \in \mathbb{Z}$ (orientation d'un objet ou de sa dynamique) de celles de la forme $[a, b], (a, b) \in \mathbb{R}^2$ (distance, échelle d'un objet). Nous pouvons définir celles de la première catégorie (qui nécessitent une invariance modulo 2π et donc une modification des propriétés de μ_1) de la manière suivante :

Définition 2.2 Soit $\mathcal{O} = [I_1 + 2i\pi, I_1 + 2(i+1)\pi], I_1 \in \mathbb{R}, i \in \mathbb{Z}$ le domaine de la partition. Soit $m_1 < m_2 < \cdots < m_{K+1}$ les K + 1 nœuds fixés de \mathcal{O} , tels que $m_1 = I_1 + 2i\pi$ et $m_{K+1} = I_1 + 2(i+1)\pi$, avec $K + 1 \ge 2$, et $\forall k \ne K + 1, m_{k+1} - m_k = h = \text{constante.}$ Une famille de K ensembles flous $\{\mathcal{A}_k\}_{k=1}^K$ de fonctions d'appartenance $\{\mu_k\}_{k=1}^K$ est une partition uniforme floue forte décrivant une relation spatiale angulaire si les ensembles flous $\{\mathcal{A}_k\}_{k=1}^K$ satisfont les propriétés suivantes :

- 1. $\forall k \in \{1, \ldots, K\}, \mu_k(m_k) = 1$,
- 2. $\forall k \in \{2, \dots, K\}, si \ x \notin [m_{k-1}, m_{k+1}], \mu_k(x) = 0,$ et $si \ x \notin [m_1, m_2] \cup [m_K, m_{K+1}], \mu_1(x) = 0,$

- 3. $\forall k \in \{2, ..., K\}, \mu_k(x) \text{ a une croissance monotone sur } [m_{k-1}, m_k] \text{ et une décroissance monotone sur } [m_k, m_{k+1}], \mu_1(x) \text{ a une croissance monotone sur } [m_K, m_{K+1}] \text{ et une décroissance monotone sur } [m_1, m_2],$
- 4. $\forall x \in \mathcal{O}, \sum_{k=1}^{K} \mu_k(x) = 1,$
- 5. $\forall k \in \{2, \dots, K\}, \forall x \in [0, h], \mu_k(m_k x) = \mu_k(m_k + x),$
- 6. $\forall k \in \{2, \dots, K\}, \forall x \in [m_k, m_{k+1}], \mu_k(x) = \mu_{k-1}(x-h),$
- 7. $\forall k \in \{2, \dots, K-1\}, \forall x \in [m_{k-1}, m_k], \mu_k(x) = \mu_{k+1}(x+h), \\ \forall x \in [m_{K-1}, m_K], \mu_K(x) = \mu_1(x+h),$
- 8. $\forall (h,k) \in \{1, \dots, K\}^2, \int_{\mathcal{O}} \mu_h(x) \, dx = \int_{\mathcal{O}} \mu_k(x) \, dx.$



FIG. 2.7 – Exemple de partition uniforme floue forte décrivant une relation spatiale angulaire, avec K = 8 ensembles flous.

Un exemple de famille d'ensembles flous décrivant une relation spatiale angulaire est illustré sur la figure 2.7.

Nous définissons maintenant les relations spatiales non angulaires selon :

Définition 2.3 Soit $\mathcal{O} = [I_1, I_2], (I_1, I_2) \in \mathbb{R}^2$ le domaine de la partition. Soit $m_1 < m_2 < \cdots < m_K$ les K nœuds fixés de \mathcal{O} , avec $K \ge 3$, tels que $\forall k \ne K, m_{k+1} - m_k = h = \text{constante}$. Les nœuds des extrémités sont fixés de sorte que $m_1 = I_1 + (I_2 - I_1)/2K$ et $m_K = I_2 - (I_2 - I_1)/2K$. Une famille de K ensembles flous $\{\mathcal{A}_k\}_{k=1}^K$ de fonctions d'appartenance $\{\mu_k\}_{k=1}^K$ est une partition floue décrivant une relation spatiale non angulaire si les ensembles flous $\{\mathcal{A}_k\}_{k=1}^K$ satisfont les propriétés 1 à 8 définies respectivement dans la définition 2.1 et l'équation 2.5.



FIG. 2.8 – Partition uniforme floue de la relation spatiale distance avec K = 3 ensembles flous. La fonction d'appartenance de l'ensemble flou *Moyen* est de forme trapézoïdale, de noyau 2b et de support 2(a + b).

Un exemple de famille d'ensembles flous décrivant une relation spatiale non angulaire est illustré sur la figure 2.8. La construction d'une telle famille est en pratique simple. Soit 2*b* la longueur des noyaux des fonctions qui ne se situent pas aux bords et $a \triangleq h - 2b$ la longueur de pente de chaque côté. Alors les fonctions d'appartenance trapézoïdales d'une partition floue définissant des relations spatiales de distance s'écrivent :

$$\mu_{k}(x) = \begin{cases} (x - (m_{k} - b - a))/a & \text{si } x \in]m_{k} - b - a, m_{k} - b[\\ 1 & \text{si } x \in [m_{k} - b, m_{k} + b]\\ 1 - (x - (m_{k} + b))/a & \text{si } x \in]m_{k} + b, m_{k} + b + a[\\ 0 & \text{sinon} \end{cases}$$
$$\mu_{1}(x) = \begin{cases} 1 & \text{si } x \in [I_{1}, m_{2} - b - a]\\ 1 - (x - (m_{2} - b - a))/a & \text{si } x \in]m_{2} - b - a, m_{2} - b[\\ 0 & \text{sinon} \end{cases}$$
$$\mu_{K}(x) = \begin{cases} (x - (m_{K-1} + b))/a & \text{si } x \in]m_{K-1} + b, m_{K-1} + b + a[\\ 1 & \text{si } x \in [m_{K-1} + b + a, I_{2}]\\ 0 & \text{sinon} \end{cases}$$

Notons que, les fonctions étant linéaires ou constantes par morceaux, la simulation d'un échantillon (qui sera nécessaire dans le cadre du filtre particulaire développé dans le chapitre 3) peut alors être réalisée de manière analytique par une méthode d'inversion de la fonction de répartition (sections 1.1.2.1 et 3.5.1). Enfin, nous pouvons noter que les relations spatiales définies dans cette section sont fortement contraintes, en particulier car les familles de fonctions d'appartenance doivent former une partition uniforme floue forte. Cela permet en pratique de garantir une exploration de l'espace d'état uniforme dans le processus de suivi décrit dans le chapitre 3. Toutefois, nous aurions pu imaginer intégrer des relations spatiales non contraintes, plus spécifiques à une application, à l'instar des informations de formes floues présentées dans la section 2.3.

L'ensemble flou \mathcal{Z} des événements possibles correspond ici simplement à la somme des fonctions d'appartenance, c'est-à-dire, $\forall x \in \mathcal{O}, \mu_{\mathcal{Z}}(x) = \sum_{k=1}^{K} \mu_k(x) = 1$ (voir section 2.1.4). Cela garantit en pratique une exploration uniforme de l'espace \mathcal{O} .

2.2.1.2 Cas multidimensionnel

Une extension de la définition de partition floue au cas multidimensionnel est nécessaire lorsque nous souhaitons représenter des concepts tels que la vélocité, le vecteur d'accélération ou encore les dimensions d'un objet représenté dans l'espace de l'image (2D) ou dans l'espace de l'objet (3D). En considérant R dimensions, les fonctions d'appartenance sont de la forme $\mu_{k^1,...,k^R}$, avec $\{k^r\}_{r=1}^R$ les indices définis sur chacune des dimensions. Nous notons $\mu_{k^r}^r$ la k^r -ième fonction d'appartenance définie sur la r-ième dimension. Nous définissons tout d'abord la notion de partition uniforme floue dans le cas multidimensionnel :

Définition 2.4 Soit $\mathcal{O} = [I_1^1, I_2^1] \times \cdots \times [I_1^R, I_2^R] \subset \mathbb{R}^R$ le domaine de la partition de dimension R. Soit $\forall r \in \{1, \ldots, R\}, m^r = (m_1^r, \ldots, m_{K^r}^r)$ les K^r nœuds fixés de l'intervalle $[I_1^r, I_2^r]$, tels que $m_1^r = I_1^r$ et $m_K^r = I_2^r$, avec $K^r \ge 2$, et $\forall k \ne K^r, m_{k+1}^r - m_k^r = h^r$. La famille des $\prod_{r=1}^R K^r$ ensembles flous $\{\mathcal{A}_{k^1,\ldots,k^R}\}$ définis par leurs fonctions d'appartenance $\{\mu_{k^1,\ldots,k^R}\}$ sur \mathcal{O} forme une partition uniforme floue forte multidimensionnelle si elle satisfait les propriétés suivantes :

1. $\forall (k^1, \dots, k^R) \in \{1, \dots, K^1\} \times \dots \times \{1, \dots, K^R\},\$

$$\mu_{k^1,\dots,k^R}(m_{k^1}^1,\dots,m_{k^R}^R) = 1$$

2. $\forall (k^1, \dots, k^R) \in \{1, \dots, K^1\} \times \dots \times \{1, \dots, K^R\}, \forall x \notin [m^1_{k^1-1}, m^1_{k^1+1}] \times \dots \times \{1, \dots, K^R\}$ $[m_{kR-1}^{R}, m_{kR+1}^{R}],$

 $\mu_{k^1,\dots,k^R}(x) = 0$

- 3. $\forall r \in (1, \ldots, R), \forall k^r \in \{1, \ldots, K^r\}, \mu_{k^r}^r(x)$ a une croissance monotone sur $[m_{k^r-1}^r, m_{k^r}^r]$ *et une décroissance monotone sur* $[m_{k^r}^r, m_{k^r+1}^r]$
- 4. $\forall x \in \mathcal{O}$,

7.

$$\sum_{k^{1}=1}^{K^{1}} \cdots \sum_{k^{R}=1}^{K^{R}} \mu_{k^{1},\dots,k^{R}}(x) = 1$$

- 5. $\forall (k^1, \dots, k^R) \in \{2, \dots, K^1 1\} \times \dots \times \{2, \dots, K^R 1\}, \forall x \in [0, h^1] \times \dots \times [0, h^R],$ $\mu_{k^1,\dots,k^R}(m_{k^1}^1 - x^1,\dots,m_{k^R}^R - x^R) = \mu_{k^1,\dots,k^R}(m_{k^1}^1 + x^1,\dots,m_{k^R}^R + x^R)$
- 6. $\forall (k^1, \dots, k^R) \in \{2, \dots, K^1 1\} \times \dots \times \{2, \dots, K^R 1\}, \forall x \in [m_{k^1}^1, m_{k^1 + 1}^1] \times \dots \times \{2, \dots, K^R 1\}, \forall x \in [m_{k^1}^1, m_{k^1 + 1}^1] \times \dots \times \{m_{k^1}^1, m_{k^1 + 1}^1\}$ $[m_{kR}^{R}, m_{kR+1}^{R}],$

$$\mu_{k^1,\dots,k^R}(x) = \mu_{k^1-1,\dots,k^R-1}(x-h)$$

$$\forall (k^1,\dots,k^R) \in \{2,\dots,K^1-1\} \times \dots \times \{2,\dots,K^R-1\}, \forall x \in [m^1_{k^1-1},m^1_{k^1}] \times \dots \times [m^R_{k^R-1},m^R_{k^R}],$$

$$\mu_{k^1,\dots,k^R}(x) = \mu_{k^1+1,\dots,k^R+1}(x+h)$$

Notons que toutes les variantes des propriétés 5, 6 et 7 peuvent être obtenues directement. De même, la propriété énoncée dans l'équation 2.5 s'écrit dans le cas multidimensionnel :

8. $\forall (k^1, l^1) \times \cdots \times (k^R, l^R) \in \{1, \dots, K^1\}^2 \times \cdots \times \{1, \dots, K^R\}^2$,

$$\int_{\mathcal{O}} \mu_{k^1,\dots,k^R}(x) \, dx = \int_{\mathcal{O}} \mu_{l^1,\dots,l^R}(x) \, dx \tag{2.6}$$

1)

Nous construisons ensuite la partition uniforme floue forte multidimensionnelle à partir de partitions uniformes floues fortes monodimensionnelles :

Proposition 2.1 Soit $\mathcal{O} = [I_1^1, I_2^1] \times \cdots \times [I_1^R, I_2^R] \subset \mathbb{R}^R$ le domaine de la partition de dimension R. Soit $\forall r \in \{1, ..., R\}, m^r = (m_1^r, ..., m_{K^r}^r)$ les K^r -nœuds fixés de l'intervalle $[I_1^r, I_2^r]$, tels que $m_1^r = I_1^r$ et $m_K^r = I_2^r$, avec $K^r \ge 2$, et $\forall k \ne K^r, m_{k+1}^r - m_k^r = h^r$. Soit $\forall (k^1, ..., k^R) \in \{1, ..., K^1\} \times \cdots \times \{1, ..., K^R\}, \forall x \in \mathcal{O}$, la fonction d'appartenance jointe $\mu_{k^1,...,k^R}(x) = \prod_{r=1}^R \mu_{k^r}^r(x^r).$

Si $\forall r \in \{1, ..., R\}$, la famille des ensembles flous $\{\mathcal{A}_{k^r}^r\}_{k^r=1}^{K^r}$ identifiés par leurs fonctions d'appartenance $\{\mu_{k^r}^r\}_{k^r=1}^{K^r}$ forme une partition uniforme floue forte sur le domaine $[I_1^r, I_2^r]$, alors la *famille des ensembles flous* $\{A_{k^1,\ldots,k^R}\}$ *identifiés par leurs fonctions d'appartenance* $\{\mu_{k^1,\ldots,k^R}\}$ forme une partition uniforme floue forte sur \mathcal{O} .

De même, la famille des ensembles flous $\{A_{k^1,\ldots,k^R}\}$ identifiés par leurs fonctions d'appartenance $\{\mu_{k^1,\dots,k^R} = \prod_{r=1}^R \mu_{k^r}^r(x^r)\}$ vérifie la propriété énoncée dans l'équation 2.6, lorsque $\forall r \in \{1, \dots, R\}$, les fonctions d'appartenance $\{\mu_{k^r}^r\}_{k^r=1}^{K^r}$ forment une partition uniforme floue forte sur le domaine $[I_1^r, I_2^r]$ (et vérifient ainsi la propriété définie dans l'équation 2.5). Les preuves sont données en annexe A. Les fonctions d'appartenance décrivant des relations spatiales multidimensionnelles sont donc définies simplement par $\mu_{k^1,\ldots,k^R}(x) = \prod_{r=1}^R \mu_{k^r}^r(x^r)$, avec $\mu_{k^r}^r(x^r)$ une relation spatiale monodimensionnelle de définition 2.2 ou 2.3. Ainsi pour générer un échantillon, il suffit de considérer indépendamment chacun des domaines monodimensionnels.

2.2.2 Transition

Dans le but d'introduire ces relations spatiales non seulement dans la variable d'état (la variable cachée du système) du filtre particulaire, mais aussi dans sa dynamique, nous proposons de modéliser le changement potentiel d'un état (au sens de statut) en favorisant le passage à ce même état, et, dans une moindre mesure, le changement vers un état voisin. Dans ce dernier sens, un état désigne donc un ensemble flou particulier d'un concept flou. Par état voisin, nous entendons celui representé par l'une des deux fonctions d'appartenance les plus proches de celle de l'état qui est considéré (par exemple, les voisins de *nord* sont *nord-est* et *nord-ouest*). Cette idée est appropriée lorsqu'un objet change graduellement d'état, au sens de sa dynamique. Il est donc naturel de définir la transition d'un état *h* vers un état *k* en utilisant le degré d'intersection R(k,h) contraint défini dans l'équation 2.4.

2.2.2.1 Cas monodimensionnel

Plusieurs degrés d'intersection sont définis dans la litérature [Bloch, 2005]. Le plus simple d'entre eux est certainement le suivant :

$$\Xi(\mu_k, \mu_l) = \sup_{\mathbf{o} \in \mathcal{O}} \top \left[\mu_k(\mathbf{o}), \mu_l(\mathbf{o}) \right]$$
(2.7)

Cependant, cet opérateur n'intègre pas d'information volumique et, dans le cadre du partitionnement flou fort défini, sera systématiquement égal à 0 si les patrons utilisés sont rectangulaires, et à 0.5 sinon. Nous prenons donc en compte la surface (ou le volume) de l'intersection floue et nous définissons le degré d'intersection $\Xi(\mu_k, \mu_h)$ dans le cadre des relations spatiales entre les ensembles flous μ_k et μ_h par :

$$\Xi(\mu_k,\mu_l) = \frac{\int_{\mathcal{O}} \top [\mu_k(\mathbf{o}),\mu_l(\mathbf{o})] \, d\mathbf{o}}{\min\left[\int_{\mathcal{O}} \mu_k(\mathbf{o}) \, d\mathbf{o}, \int_{\mathcal{O}} \mu_l(\mathbf{o}) \, d\mathbf{o}\right]}$$
(2.8)

En prenant $\top = \min$, le degré d'intersection $\Xi(\mu_k, \mu_l)$ est proportionnel à l'aire d'intersection entre les ensembles flous μ_k et μ_l . Dans la figure 2.8, le degré d'intersection entre l'état *Proche* et l'état *Moyen* est par exemple proportionnel à l'aire indiquée en orange. Plus précisemment, le degré d'intersection $\Xi(\mu_k, \mu_l)$ pour |k - l| = 1 est proportionnel à a/4, avec a la longueur de pente considérée. Le tableau 2.1 donne des exemples de matrices de degrés d'intersection obtenues en utilisant différentes formes de fonctions d'appartenance.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
(a)
$$\begin{bmatrix} 1 & 0.25 & 0 \\ 0.25 & 1 & 0.25 \\ 0 & 0.25 & 1 \end{bmatrix}$$
(b)
$$\begin{bmatrix} 1 & 0.125 & 0 \\ 0.125 & 1 & 0.125 \\ 0 & 0.125 & 1 \end{bmatrix}$$
(c)

TAB. 2.1 – Matrices de degrés d'intersection obtenues en considérant 3 classes du concept flou distance avec un patron central (a) rectangulaire, (b) triangulaire et (c) trapézoïdale de longueurs de noyau 2b = 2 et de support 2(b + a) = 6.

2.2.2.2 Cas multidimensionnel

Le cas multidimensionnel ne pose guère plus de difficultés, et peut être défini et calculé analytiquement en combinant de manière conjonctive les degrés d'intersection calculés sur chaque dimension :

$$\Xi(\mu_{k^1,\dots,k^R},\mu_{l^1,\dots,l^R}) = \mathop{\top}_{r \in \{1,\dots,R\}} \Xi^r(\mu_{k^r},\mu_{l^r})$$
(2.9)

avec $\Xi^r(\mu_{k^r}, \mu_{l^r})$ le degré d'intersection entre deux ensembles flous monodimensionnels, défini dans l'équation 2.7 ou 2.8.

2.3 Informations de formes floues

Nous nous intéressons maintenant à la construction d'informations de formes floues. Dans ce cadre, un ensemble flou décrit un ensemble de formes de degrés d'appartenance différents, mais regroupées sous une étiquette commune. Nous appelons posture cette étiquette, c'est-à-dire un ensemble flou caractérisant un ensemble de formes. Chaque élément des ensembles flous est donc une forme.

D'autres approches de la litérature [Pham, 2001; Pham et Zhang, 2001; Tanács et al., 2009] ont proposé une version floue d'un modèle descriptif de forme, potentiellement de haut niveau (comme une B-Spline). Toutefois, l'utilisation de modèles de description de formes mathématiques peut être contraignant, dans le sens où la généralisation de l'approche à d'autres types de formes peut être compromise. De plus, ces modèles n'intègrent généralement pas d'informations dynamiques, ce qui est nécessaire dans notre application. Comme nous le constaterons dans cette section, notre approche s'abstrait de toute modélisation mathématique, puisque les formes sont définies de manière implicite par des contraintes, en utilisant une notion de distance entre une forme candidate et une forme appartenant à l'ensemble flou considéré.

2.3.1 Mise en œuvre générique

Nous considérons que la création des ensembles flous décrivant des formes se fait par apprentissage. Chaque forme de la base d'apprentissage contient un ensemble de points de contrôle et une étiquette de classe. Les ensembles flous qui seront ensuite implicitement générés prendront en compte l'ensemble des formes de la base d'apprentissage. Le degré d'appartenance d'une forme candidate est calculé en fonction d'un degré de similarité, défini selon une distance choisie, entre celle-ci et la forme dont elle est la plus proche issue de la base de données.

Le schéma d'apprentissage et de construction des ensembles flous donné ici est général est pourrait certainement convenir à divers types de formes, rigides ou non. Soit $BD = \{(a^{(m)}, c^{(m)}), m = 1, ..., M\}$ une base de données d'éléments $a^{(m)} \in \mathbb{R}^{2P}$ de Ppoints de contrôle 2D annotés d'une étiquette $c^{(m)} \in \{1, ..., K\}$. L'algorithme d'apprentissage des fonctions floues illustré sur la figure 2.9 (en prenant pour exemple des formes de bouche) se résume par les points suivants :

- 1. Création de la base de données :
 - a) Collecte des *M* échantillons $\{(a^{(m)}, c^{(m)}), m = 1, \dots, M\}$.
 - b) Rotation de toutes les formes afin d'aligner leurs axes principaux (il est également possible d'effectuer une mise à l'échelle des formes).

- c) Répartition des échantillons par classe.
- 2. Création des fonctions floues associées aux échantillons $\{g_{a^{(m)}}\}_{m=1}^{M}$ (équation 2.2).
- 3. Création des fonctions floues de classes $\{\mu_k^r\}_{k=1}^{K^r}$ (équation 2.3).



FIG. 2.9 – Schéma de l'apprentissage des fonctions floues. Les formes prises en exemple sont des bouches. L'étape 1 correspond à la création de la base de données, l'étape 2 à la création des fonctions floues associées aux échantillons, et l'étape 3 à la création des fonctions floues de classes en utilisant une t-conorme $\perp = \max$. Dans les étapes 1 et 2, une seule occurrence de forme par classe est représentée, dans un souci de clarté. Les ensembles flous sont représentés en dimension 1, bien qu'en réalité ils soient définis sur \mathbb{R}^{2P} , avec *P* le nombre de points de contrôle.

Nous souhaiterions intégrer aux fonctions floues $g_{a^{(m)}}$ des contraintes de forme, afin d'être en mesure de définir une classe floue, mais qui reste cohérente avec les formes considérées. Pour ce faire, plusieurs stratégies peuvent être mises en œuvre, en définissant les contraintes de forme $C_{a^{(m)}}$. D'une manière générique, cette fonction floue peut s'écrire :

$$C_{a^{(m)}}(o) = \Lambda_{a,b}\left(d(o, a^{(m)})\right)$$

avec $\Lambda : \mathbb{R}^+ \to [0,1]$ une fonction floue décroissante illustrée sur la figure 2.10 et $d : \mathbb{R}^{2P} \times \mathbb{R}^{2P} \to \mathbb{R}^+$ une distance entre deux formes. Celle-ci peut être choisie en fonction de propriétés escomptées. Par exemple, une distance entre histogrammes de directions impliquerait une invariance par rotation, translation et changement d'échelle. En utilisant un patron flou $P_{a(m)}$ assez permissif, par exemple correspondant à une boule floue, et des contraintes $C_{a(m)}$ fortes, on obtient une classe étalée, tout en restreignant les formes afin que celles-ci soient cohérentes avec les formes présentes dans la base d'apprentissage. Nous obtenons ainsi une partition floue quelconque, dont les fonctions d'appartenance



FIG. 2.10 – Exemple de fonction floue $\Lambda_{a,b}$ décroissante de paramètres *a* et *b*.

possèdent un degré maximal lorsque la forme respecte totalement la contrainte imposée par la posture considérée, et un degré nul dans le cas inverse (figure 2.11).



FIG. 2.11 – Exemple de partition obtenue dans un contexte général, avec l'intégration de contraintes.

2.3.2 Application à la modélisation de la bouche

Nous nous intéressons maintenant à l'application du modèle général à la modélisation de la bouche. Les bases d'apprentissage et de test utilisées sont annotées et disponibles librement sur Internet à l'adresse http://personalpages.manchester. ac.uk/staff/timothy.f.cootes/data/talking_face/talking_face.html (figure 2.12). La forme de la bouche est décrite par P = 18 points de contrôle, répartis sur des points précis du contour de la bouche. La famille des fonctions de posture de la bouche est obtenue automatiquement par apprentissage, et trois classes ont été considérées : *fermée, ouverte* et *sourire*. Une bouche doit être ouverte pour pouvoir former un sourire. Ainsi, lorsque les lèvres inférieures et supérieures ne se touchent pas, la classe considérée est soit *ouverte*, soit *sourire*. Il aurait été possible de considérer d'autres classes, telles que *lèvres rentrées* ou encore *bouche triste*. Toutefois, le choix d'un petit nombre de classes permet d'aboutir à un modèle souple, ce qui le rend d'autant plus robuste aux variations rapides de la forme de la bouche.

Nous utilisons ici une fusion intra-classe disjonctive, avec des t-conormes ($\Phi = \max$). Cela mène à des classes de sémantique possibiliste, ce qui semble bien adapté, entre autres, aux applications d'apprentissage : la base de données représentant un échantillon effectivement observé, chacun de ses éléments est donc possible, réalisable. D'autre part, nous ne considérons pas de fusion inter-classes (donc $\mu_k = \tilde{\mu}_k$) et l'ensemble des événements possibles \mathcal{Z} a été défini en utilisant une stratégie d'union (2.1.4). Ces choix permettent de ne pas introduire de restriction sur les zones communes entre plusieurs postures, afin de rester cohérent avec la base d'apprentissage (comme par exemple lorsque l'individu entrouvre très légèrement la bouche).

2.3.2.1 Contraintes de formes

Afin de garantir une forme globale quasi identique à celle de l'échantillon $a^{(m)}$, tout en autorisant des changements d'échelle et locaux, nous définissons la distance de forme



FIG. 2.12 – Exemple d'une image annotée de la base de données. A la différence de cette image qui ne sert qu'à illustrer les points caractéristiques déjà annotés dans la base de données, les 5000 images qui la composent sont en couleur. De tous ces points, nous conservons uniquement ceux de la bouche, c'est-à-dire ceux dont l'étiquette est comprise entre 48 et 65.

par :

$$d(x,y) = \frac{1}{P} \sum_{p=1}^{P} \left| \widehat{(x^{p-1}, x^p, x^{p+1})} - \widehat{(y^{p-1}, y^p, y^{p+1})} \right|$$

avec (a, b, c) l'angle entre les vecteurs \vec{ab} et \vec{bc} , et les points $x^0 \triangleq x^P$ et $x^{P+1} \triangleq x^1$ considérés pour simplifier l'écriture de l'équation. Les changements d'échelle et locaux restent bien sûr mineurs, car ils sont contraints par la boule floue P_m considérée, qui a un support compact. En pratique, cette contrainte de forme permet donc de proposer des formes qui ont une configuration globale cohérente vis-à-vis de l'observation considérée, car elle tend à conserver les angles définis entre trois points de contrôle successifs.

2.3.2.2 Transition

Nous utilisons le degré d'intersection suivant entre les classes k et h :

$$\Xi(\mu_k,\mu_h) = \frac{\int_{\mathcal{O}} \mu_{\mathcal{Z}}(\mathbf{o}) \Upsilon\left[\mu_k(\mathbf{o}),\mu_h(\mathbf{o})\right] d\mathbf{o}}{\min\left[\int_{\mathcal{O}} \mu_{\mathcal{Z}}(\mathbf{o})\mu_k(\mathbf{o}) d\mathbf{o},\int_{\mathcal{O}} \mu_{\mathcal{Z}}(\mathbf{o})\mu_h(\mathbf{o}) d\mathbf{o}\right]}$$
(2.10)

avec $\Upsilon(a, b)$ un opérateur flou tel que $\Upsilon(a, b) = 1$ si $ab \neq 0$ et 0 sinon, et $\mathbb{1}^k(x) = \Upsilon(\mu_k(x), 1)$ la fonction indicatrice de la classe k, qui vaut 1 si $x \in \mathcal{A}_k$, 0 sinon. Le degré d'intersection défini ici accentue volontairement l'interaction entre deux classes, en considérant une contribution maximale d'une forme lorsque celle-ci possède une appartenance non nulle aux deux ensembles considérés. Cela permet de favoriser les transitions qui seront définies grâce à cet opérateur binaire. La fonction d'appartenance μ_Z à l'ensemble flou Z de tous les événements possibles permet de pondérer les calculs par les zones d'intérêt des fonctions d'appartenance, qui correspondent entre autres aux zones qui sont communes à plusieurs ensembles flous. La figure 2.13 illustre le fonctionnement du degré d'intersection défini.

Des exemples de simulation de bouche sont illustrés sur la figure 2.14, en utilisant différents paramètres de patron et de contraintes de forme, et menant ainsi à des degrés d'intersection entre les classes plus ou moins forts (tableau 2.2).



FIG. 2.13 – Deux ensembles flous quelconques sont représentés. Le degré d'intersection défini dans l'équation 2.10 est proportionnel à la surface de la zone délimitée en vert, en utilisant une stratégie d'union (max) pour définir l'ensemble flou \mathcal{Z} .



FIG. 2.14 – Simulations avec un patron (a) de support restreint, (b) de support large et (c) de support large avec contraintes de forme. Le rectangle rouge correspond au support d'un point de contrôle (P = 18).

Les résultats du tableau 2.2 montrent que les degrés d'intersection obtenus avec un patron large avec contraintes de forme sont plus élevés qu'avec un patron simplement large. Cela s'explique par le fait que certaines observations utilisées pour générer les fonctions d'appartenance ont été choisies afin d'obtenir des degrés d'intersection élevés entre les classes (en prenant deux formes proches mais d'étiquettes différentes). Ainsi les fonctions floues obtenues en utilisant des contraintes de forme, bien qu'étant plus restreintes que celles n'en utilisant pas, restent larges aux endroits où les formes sont cohérentes vis-à-vis des contraintes imposées, ce qui est notamment vrai aux intersections de plusieurs observations, et donc plus particulièrement dans les zones d'appartenance partagées par plusieurs classes.

	1	2	3			1	2	3			1	2	3	
1	[1	0	0		1	1	0.11	0.01]	1	1	0.20	0.02]
2	0	1	0	(a)	2	0.11	1	0.07	(b)	2	0.20	1	0.16	(c)
3	0	0	1		3	0.01	0.07	1		3	0.02	0.16	1	

TAB. 2.2 – Matrices de degrés d'intersection avec un patron (a) de support restreint, (b) de support large et (c) de support large avec contraintes de forme. Les lignes et les colonnes représentent les indices des ensembles flous, 1 pour la bouche *fermée*, 2 pour la bouche *ouverte* et 3 pour la bouche *sourire*.

2.4 Conclusion

Nous avons présenté un cadre unifié de représentation des relations spatiales floues et des formes floues. Plus précisément, les ensembles flous caractérisant des relations spatiales forment une partition uniforme floue forte de l'espace considéré, ce qui permet, entre autres, de faciliter leur manipulation en pratique. Les formes floues ont été définies par contraintes, et permettent ainsi de représenter n'importe quel type de formes. Nous avons appliqué ce modèle à la création de fonctions d'appartenance décrivant des postures floues de la bouche : *fermée, ouverte* et *sourire*, dont la base de données d'apprentissage est disponible librement sur Internet. La représentation formelle des relations spatiales décrites dans ce chapitre ainsi que la modélisation de formes floues sont des contributions originales de ce travail de thèse.

Les concepts qui ont été introduits dans ce chapitre faciliteront l'intégration des informations floues dans un contexte probabiliste tel que celui du filtre particulaire. Par exemple, le degré d'intersection entre deux ensembles flous permettra de directement construire la matrice de transition, donc définir la dynamique, entre deux ensembles flous considérés, que nous nommerons classes. En outre, l'introduction d'informations floues dans le suivi permettra de proposer une nouvelle modélisation de la problématique du suivi, de par l'utilisation de données structurelles ou caractéristiques de l'objet, ce qui n'a pas été fait auparavant de cette manière.
Chapitre 3

Intégration d'informations spatiales floues dans le filtre particulaire

Sommaire

3.1	Introduction d'informations spatiales floues dans un contexte probabiliste 74	
3.2	Introduction d'informations spatiales floues dans un filtre particulaire . 77	
	3.2.1 Modélisation des paramètres dans un filtre particulaire 77	
	3.2.2 Modélisation des paramètres dans un contexte flou	
	3.2.3 Modèle à sauts	
3.3	Algorithme de suivi	
3.4	Fonction d'importance	
	3.4.1 Modélisation de la fonction d'importance	
	3.4.2 Complexité et coût de calcul	
	3.4.3 Conclusion	
3.5	Modélisation d'informations spatiales floues en pratique	
	3.5.1 Modélisation des relations spatiales	
	3.5.2 Modélisation d'informations spatiales générales	
3.6	Approximation	
3.7	Conclusion	

Dans le chapitre précédent, nous avons modélisé des informations spatiales floues qui peuvent être de formes ou de structure. L'objectif de ce chapitre est d'intégrer ces informations floues dans un contexte probabiliste, et plus particulièrement dans le filtre particulaire, en modélisant de manière originale les paramètres dynamiques par un modèle flou souple, et non par un modèle physique souvent difficile à déterminer. Les informations spatiales floues seront donc intégrées dans l'étape de prédiction, dans le but d'améliorer la propagation des particules. Comme nous le constaterons dans le chapitre applicatif 4, cela permet en particulier de rendre le suivi robuste aux changements brusques de la dynamique de l'objet, et par la modélisation souple, de représenter simplement des informations complexes, telles que les formes.

L'intégration du flou dans le filtre particulaire a été peu abordé jusqu'à présent. Les travaux existants traitent essentiellement de la fusion d'observations par des opérateurs de fusion floue [Faux et Luthon, 2006; Garcia et al., 2003; Kim et al., 2007]. [Cai et Lin, 2008] proposent d'adapter automatiquement le nombre de particules en utilisant des

règles de logique floue. Un travail conséquent a été effectué par [Wu et al., 2008] où les auteurs proposent d'approcher des fonctions floues par un algorithme de filtre particulaire. Dans le cadre d'un algorithme de suivi par modèles multiples (MMPF, section 1.3.2), [Kamel et Badawy, 2005] proposent de sélectionner le modèle dynamique, ainsi que ses paramètres, grâce à des règles logiques floues. Bien que la réalisation et l'utilisation sont différentes, nos travaux se rapprochent dans l'esprit de ceux effectués par [Duan et al., 2006]. Les auteurs y proposent de contraindre l'espace d'état par l'utilisation de sousensembles flous, qui décrivent une connaissance du domaine, et appliquent cette méthodologie dans le cadre du diagnostic de pannes de capteurs de robots mobiles.

Ces divers travaux intègrent une information floue dans le filtre particulaire. Toutefois, les différences avec notre contribution exposée dans ce chapitre sont nombreuses. Premièrement, à la différence des approches précédemment citées, l'information floue que nous intégrons est de nature spatiale, et permet de décrire l'évolution d'un objet au cours du temps. Formellement, elle est intégrée par un modèle hybride continu / discret et sous la forme de paramètres dynamiques inconnus du système, ce qui constitue également une originalité. Cela signifie que son but est de décrire, par des informations spatiales floues, la dynamique du système, et donc de proposer une simulation des particules pertinente pendant la phrase de prédiction. Cela se démarque également de la plupart des autres approches qui intégrent l'information floue pendant la mise à jour des poids, via la vraisemblance. Enfin, nous proposons également une fonction d'importance originale, qui est adaptée au type de modélisation floue que nous proposons.

Dans notre modélisation, chaque concept flou prend ses valeurs dans un ensemble de valeurs linguistiques, représentées par des ensembles flous. La sélection de l'ensemble flou adéquat peut être vue comme un modèle à sauts, dont la matrice de transition dépend des degrés d'intersection flous étudiés dans le chapitre 2. Enfin, l'ensemble des valeurs désignées par les fonctions floues forme l'espace des paramètres dynamiques du modèle.

Ce chapitre est organisé de la manière suivante. Nous détaillerons tout d'abord une méthodologie simple pour intégrer des ensembles flous dans un contexte probabiliste (section 3.1) avant de le faire dans le filtre particulaire dans la section 3.2. L'algorithme qui en résulte est présenté dans la section 3.3. Nous proposerons ensuite une fonction d'importance dédiée au modèle que nous présentons (section 3.4). La section 3.5 traite de l'intégration des informations spatiales floues en pratique, alors que la dernière section 3.6 détaille une phase d'approximation parfois nécessaire, comme dans le cas des formes de bouche présentées dans la section 2.3 du chapitre précédent.

3.1 Introduction d'informations spatiales floues dans un contexte probabiliste

Nous avons défini dans le chapitre 2 une manière de représenter des informations spatiales floues, de nature structurelle ou de forme. Afin de les utiliser dans un filtre particulaire, par nature probabiliste, il est nécessaire de proposer une structure formelle permettant de manipuler à la fois les deux théories, qui modélisent des informations de sémantiques bien distinctes. Là où une approche probabiliste évalue le caractère probable d'un événement, une approche floue permet de représenter des données connues, mais de manière imprécise. Bien que d'autres solutions soient certainement possibles, par exemple en passant par des distributions de possibilités [Wu et al., 2008], nous avons

choisi d'utiliser le formalisme initialement proposé par [Zadeh, 1968], dont l'idée générale consiste à considérer les ensembles flous comme des événements flous, et à calculer leurs probabilités par rapport à une certaine mesure de probabilité \mathbb{P} . Ce formalisme possède les avantages d'être simple à mettre en œuvre et de combiner à la fois l'information floue et l'information probabiliste. Cela permet d'introduire l'approche proposée comme une extension possible au formalisme classique, dans laquelle l'information pertinente des relations spatiales est traduite par une approche floue, sans pour autant abandonner le formalisme probabiliste, qui traduit une information pauvre (uniforme).

Passons maintenant à la formalisation d'événements flous dans un contexte probabiliste. Soit le triplet $(\mathbb{R}^n, \mathbb{B}, \mathbb{P})$, avec \mathbb{B} la tribu de Borel et \mathbb{P} une mesure de probabilité sur \mathbb{R}^n . L'espace euclidien est considéré ici dans un souci de simplicité. Soit \mathcal{A} un événement flou défini par une fonction d'appartenance Borel-mesurable $\mu_{\mathcal{A}} : \mathbb{R}^n \to [0, 1]$. Selon [Zadeh, 1968], la probabilité d'un événement flou est définie comme :

$$\mathbb{P}(\mathcal{A}) = \int_{\mathbb{R}^n} \mu_{\mathcal{A}}(\mathbf{x}) d\mathbb{P}$$

= $\mathbb{E}[\mu_{\mathcal{A}}]$ (3.1)

D'autres approches visant à définir une théorie de la mesure sur des ensembles flous ont depuis lors été étudiées (voir par exemple [Yager, 2002; Klement et Weber, 1999] pour une bibliographie étendue). Zadeh lui-même a étendu le principe de mesure de probabilité sur des événements flous aux mesures floues sur des événements flous [Zadeh, 1978]. Parallèlement et indépendemment à ce dernier travail, [Sugeno, 1974] a proposé de définir des mesures floues sur des événements flous en remplaçant l'intégrale de Lebesgue par celle de Sugeno. Il s'avérera par la suite que ces travaux peuvent se généraliser, tout en restant cohérents avec les notions originelles introduites par Zadeh et Sugeno. Ainsi, dans [Grabisch et Sugeno, 1990; Grabisch et al., 1992], les auteurs utilisent la classe plus générale des intégrales de t-conormes floues afin de définir des mesures floues sur un événement flou, et ce de manière à ce que la définition de mesure floue soit cohérente avec celle de Zadeh. Les définitions de [Zadeh, 1968, 1978] ont en fait donné lieu à de nombreux travaux sur les mesures floues. La plupart des modèles restent cohérents avec les définitions initiales, à l'instar de [Navara, 2005], qui rapproche les mesures sur des tribus [Butnariu et Klement, 1993] de mesures sur des MV-algèbres [Riečan et Mundici, 2002] utilisées dans le cadre de la définition de mesure floue, et qui restent conformes aux propositions faites par Zadeh. Toutefois, les concepts dont nous avons besoin dans cette section étant de base, nous suivrons ici le modèle d'extension intuitif et originel proposé par [Zadeh, 1968], puisqu'il suffit aux concepts simples que nous souhaitons exposer. Enumérons maintenant certaines définitions liées aux ensembles flous :

- 1. Inclusion : $\mathcal{A} \subset \mathcal{B} \Leftrightarrow \mu_{\mathcal{A}}(\mathbf{x}) \leq \mu_{\mathcal{B}}(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n$,
- 2. Egalité : $\mathcal{A} = \mathcal{B} \Leftrightarrow \mu_{\mathcal{A}}(\mathbf{x}) = \mu_{\mathcal{B}}(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n$,
- 3. Complémentarité : $\mathcal{A}^C \Leftrightarrow \mu_{\mathcal{A}^C}(\mathbf{x}) = 1 \mu_{\mathcal{A}}(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n$,
- 4. Conjonction : $\mathcal{A} \top \mathcal{B} \Leftrightarrow \mu_{\mathcal{A} \top \mathcal{B}}(\mathbf{x}) = \top (\mu_{\mathcal{A}}(\mathbf{x}), \mu_{\mathcal{B}}(\mathbf{x})), \forall \mathbf{x} \in \mathbb{R}^n$,
- 5. Disjonction : $\mathcal{A} \perp \mathcal{B} \Leftrightarrow \mu_{\mathcal{A} \perp \mathcal{B}}(\mathbf{x}) = \perp(\mu_{\mathcal{A}}(\mathbf{x}), \mu_{\mathcal{B}}(\mathbf{x})), \forall \mathbf{x} \in \mathbb{R}^n$,

avec \top et \perp des opérateurs respectivement de t-norme (conjonction floue) et t-conorme (disjonction floue) [Dubois et Prade, 1980]. D'autres formes sont possibles afin de définir

la complémentarité. Parmi les conjonctions et disjonctions floues, nous pouvons énumérer plusieurs opérateurs particuliers, définis par les concepts d'intersection, de produit, d'union et de somme :

- 4. Conjonction :
 - Intersection floue : $\mathcal{A} \cap \mathcal{B} \Leftrightarrow \mu_{\mathcal{A} \cap \mathcal{B}}(\mathbf{x}) = \min(\mu_{\mathcal{A}}(\mathbf{x}), \mu_{\mathcal{B}}(\mathbf{x})), \forall \mathbf{x} \in \mathbb{R}^n$,
 - Produit : $\mathcal{AB} \Leftrightarrow \mu_{\mathcal{AB}}(\mathbf{x}) = \mu_{\mathcal{A}}(\mathbf{x})\mu_{\mathcal{B}}(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n$,
- 5. Disjonction :
 - Union floue : $\mathcal{A} \cup \mathcal{B} \Leftrightarrow \mu_{\mathcal{A} \cup \mathcal{B}}(\mathbf{x}) = \max(\mu_{\mathcal{A}}(\mathbf{x}), \mu_{\mathcal{B}}(\mathbf{x})), \forall \mathbf{x} \in \mathbb{R}^n$,
 - Somme : $\mathcal{A} \oplus \mathcal{B} \Leftrightarrow \mu_{\mathcal{A} \oplus \mathcal{B}}(\mathbf{x}) = \mu_{\mathcal{A}}(\mathbf{x}) + \mu_{\mathcal{B}}(\mathbf{x}) \mu_{\mathcal{A}}(\mathbf{x})\mu_{\mathcal{B}}(\mathbf{x}), \forall \mathbf{x} \in \mathbb{R}^n.$

Pour la conjonction, nous utilisons la t-norme produit afin de rester cohérents avec un cadre probabiliste. Cette cohérence se traduit par exemple lorsque nous souhaitons définir l'indépendance entre deux événements flous \mathcal{A} et \mathcal{B} . L'utilisation d'une conjonction produit permet alors de définir l'indépendance d'événements flous de la même manière que s'ils n'étaient pas flous. Formellement, deux événements flous \mathcal{A} et \mathcal{B} sont déclarés indépendants s'ils satisfont :

$$\mathbb{P}(\mathcal{AB}) = \mathbb{P}(\mathcal{A}) \mathbb{P}(\mathcal{B})$$
(3.2)

où $\mathbb{P}(\mathcal{AB})$ est la probabilité de l'événement \mathcal{AB} (produit de \mathcal{A} et \mathcal{B}) défini par la fonction d'appartenance $\mu_{\mathcal{A}}\mu_{\mathcal{B}}$. En utilisant à nouveau la t-norme produit, l'expression de la probabilité conditionnelle d'un événement flou \mathcal{A} de fonction d'appartenance $\mu_{\mathcal{A}}$ sachant un événement flou \mathcal{B} de fonction d'appartenance $\mu_{\mathcal{B}}$ s'écrit :

$$\mathbb{P}(\mathcal{A}|\mathcal{B}) = \frac{\mathbb{P}(\mathcal{A}\mathcal{B})}{\mathbb{P}(\mathcal{B})}$$
$$= \frac{\int_{\mathbb{R}^n} \mu_{\mathcal{A}}(\mathbf{x}) \mu_{\mathcal{B}}(\mathbf{x}) d\mathbb{P}}{\int_{\mathbb{R}^n} \mu_{\mathcal{B}}(\mathbf{x}) d\mathbb{P}}$$
(3.3)

L'espérance sur un ensemble flou A de fonction d'appartenance μ_A peut également être définie selon :

$$\mathbb{E}_{\mu_{\mathcal{A}}}[\phi(\mathbf{x})] = \frac{1}{\mathbb{P}(\mathcal{A})} \int_{\mathbb{R}^n} \phi(\mathbf{x}) \mu_{\mathcal{A}}(\mathbf{x}) \, d\mathbb{P}$$
(3.4)

pour toute fonction ϕ continue bornée sur \mathbb{R}^n et avec $\mathbb{P}(\mathcal{A})$ un facteur de normalisation. De même, la variance s'écrit :

$$\operatorname{var}_{\mu_{\mathcal{A}}}[\phi(\mathbf{x})] = \frac{1}{\mathbb{P}(\mathcal{A})} \int_{\mathbb{R}^n} (\phi(\mathbf{x}) - \mathbb{E}_{\mu_{\mathcal{A}}}[\phi(\mathbf{x})])^2 \,\mu_{\mathcal{A}}(\mathbf{x}) \, d\mathbb{P}$$
(3.5)

Les expressions définies dans les équations 3.4 et 3.5 sont des extensions naturelles des opérateurs d'espérance et de variance, sur un ensemble flou, et en cohérence avec le formalisme décrit par [Zadeh, 1968] (équation 3.1). Elles servent à calculer des caractéristiques sur la probabilité considérée \mathbb{P} , sur un événement flou particulier, ici noté \mathcal{A} . Par exemple, en prenant $\mathcal{A} = \mathbb{R}^n$, avec $\mu_{\mathcal{A}}(\mathbf{x}) = \mu_{\mathbb{R}^n}(\mathbf{x}) = 1, \forall \mathbf{x} \in \mathbb{R}^n$, on retrouve les formulations usuelles de l'espérance mathématique et de la variance.

3.2 Introduction d'informations spatiales floues dans un filtre particulaire

3.2.1 Modélisation des paramètres dans un filtre particulaire

Comme nous l'avons vu dans la section 1.2, le problème classique de filtrage peut être décrit par :

$$\begin{cases} \mathbf{x}_t = f_t(\mathbf{x}_{t-1}, \mathbf{v}_t) \\ \mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{w}_t) \end{cases}$$
(3.6)

avec $\mathbf{x}_t \in \mathcal{X}$ la variable d'état cachée du système. Bien souvent, dans les applications de suivi d'objets dans des séquences d'images, \mathbf{x}_t contient la position dans le repère spatial ainsi que ses paramètres, dynamiques, tels que la vélocité, l'accélération, et/ou de formes, dont l'évolution au cours du temps est définie par le modèle dynamique physique $f_t(\mathbf{x}_{t-1}, \mathbf{v}_t)$. La séparation entre les variables de position et de dynamique n'est donc pas effectuée explicitement. Cependant, en vue d'introduire explicitement les informations spatiales floues, nous commençons par faire cette distinction, en introduisant les paramètres inconnus $\mathbf{r}_t \in \mathcal{R}$ de l'état \mathbf{x}_t , avec \mathcal{R} l'espace des paramètres. Le vecteur de paramètres est composé de R composantes indépendantes, $\mathbf{r}_t = (\mathbf{r}_t^1, \dots, \mathbf{r}_t^R)$, dont chacune est typiquement liée à une dynamique de marche aléatoire continue, ce qui mène au système suivant :

$$\begin{cases} \mathbf{x}_{t} = f_{t}(\mathbf{x}_{t-1}, \mathbf{r}_{t-1}, \mathbf{v}_{t}) \\ \mathbf{r}_{t}^{r} = \mathbf{r}_{t-1}^{r} + \mathbf{s}_{t}^{r} \quad \forall r \in \{1, \dots, R\} \\ \mathbf{y}_{t} = h_{t}(\mathbf{x}_{t}, \mathbf{w}_{t}) \end{cases}$$
(3.7)

avec $\mathbf{s}_t = (\mathbf{s}_t^r)_{r=1,...,R}$ des bruits gaussiens indépendants. En pratique, l'indépendance et la dynamique de marche aléatoire continue sont classiques, et permettent de gérer la plupart des problèmes de suivi. Le sytème 3.7 est équivalent à celui présenté dans l'équation 3.6, à la différence que les paramètres dynamiques sont explicités. Du système 3.7, nous opérons une modification dans le but d'introduire les informations spatiales floues :

$$\begin{cases} \mathbf{x}_t = f_t(\mathbf{x}_{t-1}, \mathbf{r}_t, \mathbf{c}_t, \mathbf{v}_t) \\ \mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{c}_t, \mathbf{w}_t) \end{cases}$$
(3.8)

avec c_t un vecteur aléatoire discret, qui indique à quels ensembles flous les composantes du vecteur de paramètres r_t appartiennent. La matrice de transition de ce vecteur est définie par les ensembles flous, et sa définition complète sera donnée dans la section 3.2.3. Ainsi, dans le modèle flou que nous proposons, la dynamique de marche aléatoire continue des paramètres n'est pas modélisée. A la place, les paramètres sont liés à un ensemble flou, restreignant ainsi son domaine de définition. Parallèlement à cela, la loi des paramètres est uniforme, ce qui permet d'effectuer une recherche dans l'espace des paramètres plus exhaustive. La combinaison de la loi uniforme et de la restriction par un ensemble flou (qui sera également estimé) permet d'obtenir un modèle dynamique souple et efficace. Une dernière différence entre les systèmes 3.7 et 3.8 concerne l'utilisation du vecteur de paramètres dans la fonction de transition f_t . Dans le système flou, nous considérons le paramètre à l'instant t, qui est défini par la valeur de c_t , et cela induit en pratique que l'ordre de simulation des variables commence par le vecteur aléatoire discret, puis le vecteur de paramètres avant de traiter x_t . Cette nuance n'est pas sans importance, puisque cela signifie que pour construire l'état du système x_t , la prédiction au temps tdes paramètres est utilisée, et ainsi la mise à jour des poids intervenant dans l'algorithme du filtre particulaire prend en compte coinjointement ces deux hypothèses issues de l'instant courant. A l'inverse, dans le système 3.7, les paramètres estimés ne seront évalués qu'à l'instant suivant, via la simulation de x_{t+1} , ce qui peut être moins robuste en pratique. La raison de ce dernier choix est liée à la définition formelle de la dynamique, en tant que modèle physique, de l'objet à suivre. A ce titre, le modèle 3.7 est cohérent, mais ne permet pas de générer le vecteur de paramètres avant x_t , ce qui semble bien indiqué dans le modèle flou que nous proposons. La différence entre les deux systèmes 3.7 et 3.8 est illustrée sur la figure 3.1.



FIG. 3.1 – Illustrations des modèles décrits dans les systèmes (a) 3.7 et (b) 3.8. Dans (a), les paramètres dynamiques estimés à t - 1 sont utilisés pour décrire la dynamique de x_t . Dans (b), les indices des ensembles flous permettent d'estimer les paramètres à l'instant courant.

3.2.2 Modélisation des paramètres dans un contexte flou

Nous modélisons maintenant les paramètres dynamiques dans notre modèle flou. Soit **r** la variable aléatoire continue représentant l'état du paramètre à valeurs dans l'espace \mathcal{R} et $d\mathbf{r}$ un événement flou de support infinitésimal. La fonction d'appartenance $\mu_{d\mathbf{r}}(\mathbf{r})$ est définie par $\mu_{d\mathbf{r}}(\mathbf{r}) = 1$ si $\mathbf{r} \in [\mathbf{r} - \frac{d\mathbf{r}}{2}, \mathbf{r} + \frac{d\mathbf{r}}{2}]$, 0 sinon. Nous considérons une collection de K ensembles flous $\{\mathcal{A}_k\}_{k=1}^K$ définis par les fonctions d'appartenance $\{\mu_k\}_{k=1}^K$, qui sont les valeurs linguistiques possibles d'un certain concept flou. Notre but est de définir les distributions du paramètre dynamique **r** sachant l'ensemble des événements flous considérés $\{\mathcal{A}_k\}_{k=1}^K$.

Tout d'abord, nous définissons l'espace des paramètres comme l'ensemble binaire des valeurs possibles :

$$\mathcal{R} = \{ \mathbf{r} \in \Omega; \text{ tel que } \mu_{\mathcal{Z}}(\mathbf{r}) > 0 \}$$
(3.9)

avec Ω l'espace des paramètres non contraints et \mathcal{Z} l'ensemble flou de tous les événements possibles (section 2.1.4). Afin d'introduire une information floue dans le filtre particulaire, nous faisons maintenant l'hypothèse que nous n'avons pas d'information *a priori* spécifique à propos de la densité de probabilité des paramètres. Nous modélisons ainsi $p(\mathbf{r})$ comme une densité de probabilité uniforme, et par conséquent, $\mathbb{P}(\mathbf{r} \in d\mathbf{r})$ comme une distribution uniforme. L'idée du modèle est de compenser le manque d'information probabiliste par un ensemble d'événements flous bien choisis, de manière à restreindre, par le flou et les informations qu'il modélise, l'exploration de l'espace des paramètres. La probabilité de l'événement \mathcal{Z} est donc :

$$\mathbb{P}(\mathcal{Z}) = \int_{\mathcal{R}} \mu_{\mathcal{Z}}(\mathbf{r}) \, d\mathbb{P} \propto \int_{\mathcal{R}} \mu_{\mathcal{Z}}(\mathbf{r}) \, d\mathbf{r}$$
(3.10)

Nous conditionnons maintenant notre modèle par l'ensemble flou des événements possibles \mathcal{Z} . Cela restreint l'ensemble des valeurs possibles des paramètres **r**. Considérant l'équation 3.3, la distribution conditionnelle $\mathbb{P}(\mathbf{r} \in d\mathbf{r}|\mathcal{Z})$ est définie par :

$$\mathbb{P}(\mathbf{r} \in d\mathbf{r} | \mathcal{Z}) = \frac{\mathbb{P}(d\mathbf{r} \mathcal{Z})}{\mathbb{P}(\mathcal{Z})} \\
= \frac{\int_{\mathcal{R}} \mu_{d\mathbf{r}}(\mathbf{r}') \mu_{\mathcal{Z}}(\mathbf{r}') d\mathbf{r}'}{\int_{\mathcal{R}} \mu_{\mathcal{Z}}(\mathbf{r}') d\mathbf{r}'} \\
= \frac{d\mathbf{r} \mu_{\mathcal{Z}}(\mathbf{r})}{\int_{\mathcal{R}} \mu_{\mathcal{Z}}(\mathbf{r}') d\mathbf{r}'}$$
(3.11)

La probabilité $\mathbb{P}(\mathbf{r} \in d\mathbf{r}|\mathcal{Z})$ est proportionnelle à $d\mathbf{r}\mu_{\mathcal{Z}}(\mathbf{r})$, qui conditionne ainsi la manière dont les paramètres dynamiques seront traités.

3.2.3 Modèle à sauts

Les ensembles flous sont intégrés dans le filtre particulaire comme des partitions floues de l'espace des paramètres de l'état inconnu du système. Le modèle flou effectue ainsi des sauts entre les ensembles flous prédéfinis. Nous modélisons ce phénomène en intégrant à l'estimation une variable aléatoire discrète qui indique à quel ensemble flou le paramètre est susceptible d'appartenir à un instant donné. Ainsi, ce modèle peut être vu comme un modèle dynamique à sauts, comme ceux proposés dans [Isard et Blake, 1998b; Pavlovic et al., 2000] ou dans les approches avec modèles multiples [McGinnity et Irwin, 2000; Karlsson et Bergman, 2000]. Cependant, la différence principale entre ces modèles et notre approche concerne la manière dont la variable de saut est utilisée. Dans les méthodes existantes, elle permet généralement de passer d'un modèle dynamique à un autre, tandis que dans notre approche, le saut est effectué dans l'espace des paramètres et permet ainsi de passer d'un modèle flou, caractérisé par un ensemble flou, à un autre. De plus, comme nous le constaterons, l'approche floue permet de déterminer le saut automatiquement, sans fixer de matrice de transition à la main (selon une approche *ad hoc*), ce qui est plus graduel, et mène à plus de souplesse.

Soit c une variable aléatoire discrète à valeurs dans $\{1, \ldots, K\}$ qui indique l'indice de l'ensemble flou considéré. Nous fixons la distribution conditionnelle de c sachant les paramètres dynamiques r et \mathcal{Z} comme proportionnelle au rapport entre le degré d'appartenance de r au $k^{ième}$ ensemble flou et son degré d'appartenance à l'ensemble flou des événements possibles \mathcal{Z} . Ainsi cette probabilité sera d'autant plus forte lorsque l'ensemble des possibles \mathcal{Z} au point r sera majoritairement décrit par le k-ième ensemble flou \mathcal{A}_k . La distribution conditionnelle est finalement obtenue en utilisant un terme de normalisation qui correspond au ratio de la somme des degrés d'appartenance des ensembles flous par celui de Z :

$$\mathbb{P}(\mathbf{c} = k | \mathbf{r}, \mathcal{Z}) = \frac{\mu_k(\mathbf{r})}{\mu_{\mathcal{Z}}(\mathbf{r})} / \frac{\sum_{h=1}^{K} \mu_h(\mathbf{r})}{\mu_{\mathcal{Z}}(\mathbf{r})}$$
$$= \frac{\mu_k(\mathbf{r})}{\sum_{h=1}^{K} \mu_h(\mathbf{r})}$$
(3.12)

A partir des équations 3.11 et 3.12, la distribution conditionnelle de c sachant l'événement flou \mathcal{Z} s'écrit :

$$\mathbb{P}(\mathbf{c} = k|\mathcal{Z}) = \int_{\mathcal{R}} \mathbb{P}(\mathbf{c} = k|\mathbf{r}, \mathcal{Z}) \mathbb{P}(\mathbf{r}|\mathcal{Z}) d\mathbf{r}$$
$$= \int_{\mathcal{R}} \frac{\mu_k(\mathbf{r})}{\sum_{h=1}^{K} \mu_h(\mathbf{r})} \frac{\mu_{\mathcal{Z}}(\mathbf{r})}{\int_{\mathcal{R}} \mu_{\mathcal{Z}}(\mathbf{r}') d\mathbf{r}'} d\mathbf{r}$$
(3.13)

Nous obtenons finalement la distribution conditionnelle de r sachant c et Z:

$$\mathbb{P}(\mathbf{r} \in d\mathbf{r} | \mathbf{c} = k, \mathcal{Z}) = \frac{\mathbb{P}(\mathbf{c} = k | \mathbf{r}, \mathcal{Z}) \mathbb{P}(\mathbf{r} \in d\mathbf{r} | \mathcal{Z})}{\mathbb{P}(\mathbf{c} = k | \mathcal{Z})} \\
= \frac{d\mathbf{r}\mu_k(\mathbf{r})\mu_{\mathcal{Z}}(\mathbf{r})}{\sum_{h=1}^{K} \mu_h(\mathbf{r})} / \int_{\mathcal{R}} \frac{\mu_k(\mathbf{r}')\mu_{\mathcal{Z}}(\mathbf{r}')}{\sum_{h=1}^{K} \mu_h(\mathbf{r}')} d\mathbf{r}'$$
(3.14)

Considérant c_t comme un processus markovien, avec t représentant le temps, il est alors possible de définir la transition de c_{t-1} vers c_t par un degré d'intersection entre les ensembles flous. En effet, deux classes proches ont des sémantiques proches, et nous considérons que la transition d'une classe vers une autre est graduelle, c'est pourquoi la matrice de transition est définie proportionnellement au degré d'intersection défini dans l'équation 2.4 :

$$\mathbb{P}(\mathbf{c}_t = k | \mathbf{c}_{t-1} = h, \mathcal{Z}) = \frac{R(k, h)}{\sum_{j=1}^{K} R(k, j)}$$
(3.15)

Ainsi, la transition des variables \mathbf{c}_t se déduit automatiquement de l'expression des fonctions floues, et utilise l'idée que la probabilité de passer d'une classe h vers une classe k est liée au degré d'intersection de leurs fonctions d'appartenance. Notons que \mathbf{c}_t aura toujours une probabilité maximale de rester dans le même état puisque $R(k,k) = \Xi^{kk}(\mu_k,\mu_k) \triangleq 1$ si les μ_k sont normalisées (c'est-à-dire max $\mu_k(\mathbf{r}) = 1$).

3.3 Algorithme de suivi

Soit $\mathbf{r}_t = (\mathbf{r}_t^1, \dots, \mathbf{r}_t^R)$ le vecteur de paramètres, ses composantes étant indépendantes entre elles. Chaque paramètre \mathbf{r}_t^r est associé à un ensemble de fonctions floues $\{\mu_k^r\}_{k=1}^{K^r}$ et est ainsi défini sur l'espace associé des situations possibles (équation 3.9). Nous considérons aussi l'ensemble des événements possibles $\mathcal{Z} = \{\mathcal{Z}^1, \dots, \mathcal{Z}^R\}$ définis par les fonctions d'appartenance floues $\{\mu_z^1, \dots, \mu_z^R\}$.

Soit $\mathbf{c}_t = (\mathbf{c}_t^1, \dots, \mathbf{c}_t^R)$ le vecteur désignant à quels ensembles flous les paramètres $(\mathbf{r}_t^1, \dots, \mathbf{r}_t^R)$ sont les plus susceptibles d'être associés. En considérant ses composantes indépendantes entre elles, nous obtenons la transition *a priori* de \mathbf{c}_t :

$$\mathbb{P}(\mathbf{c}_{t} = (k^{1}, \dots, k^{R}) | \mathbf{c}_{t-1} = (h^{1}, \dots, h^{R}), \mathcal{Z}) = \prod_{r=1}^{R} \mathbb{P}(\mathbf{c}_{t}^{r} = k^{r} | \mathbf{c}_{t-1}^{r} = h^{r}, \mathcal{Z}^{r})$$
(3.16)

avec $\mathbb{P}(\mathbf{c}_t^r = k^r | \mathbf{c}_{t-1}^r = h^r, \mathcal{Z}^r)$ la matrice de transition décrite dans l'équation 3.15. Nous modélisons le vecteur de paramètres de manière à ce que \mathbf{r}_t soit conditionnellement indépendant de \mathbf{r}_{t-1} sachant \mathbf{c}_t , ce qui mène à :

$$\mathbb{P}(\mathbf{r}_t \in d\mathbf{r}_t | \mathbf{c}_t = (k^1, \dots, k^R), \mathbf{r}_{t-1}, \mathcal{Z}) = \prod_{r=1}^R \mathbb{P}(\mathbf{r}_t^r \in d\mathbf{r}_t^r | \mathbf{c}_t^r = k^r, \mathcal{Z}^r)$$
(3.17)

avec $\mathbb{P}(d\mathbf{r}_t^r | \mathbf{c}_t^r = k^r, \mathcal{Z}^r)$ la distribution introduite dans l'équation 3.14. Le modèle graphique final est illustré sur la figure 3.2.



FIG. 3.2 – Modèle graphique de l'approche proposée. Les vecteurs aléatoires situés audessus des pointillés sont cachés, ou latents ; seules les observations sont visibles. Conditionnellement à c_t , le vecteur de paramètres r_t est indépendant de r_{t-1} . Notons que notre modèle autorise la dépendance des indices des ensembles flous c_t , il est ainsi possible de considérer des versions dédiées de la fonction de vraisemblance, c'est-à-dire une par classe, comme nous le verrons dans la mise une œuvre du filtre (chapitre 4).

La loi de filtrage *a posteriori* de la trajectoire, conditionnée par l'ensemble des événements possibles \mathcal{Z} , peut être approchée par l'algorithme 3.1, dans lequel nous avons posé $\mathbf{s}_t^{(n)} = (\mathbf{x}_t^{(n)}, \mathbf{r}_t^{(n)}, \mathbf{c}_t^{(n)})^T$ afin d'alléger les notations. L'estimateur utilisé est celui de l'espérance de Monte-Carlo (section 1.2.3).

3.4 Fonction d'importance

Un des enjeux dans le filtre particulaire est de définir une fonction d'importance pertinente, c'est-à-dire qui dépende à la fois de la trajectoire de l'état et des observations. Dans la litérature, celle-ci est le plus généralement définie par l'application, et donc fixée de manière *ad hoc*. Cependant, comme nous l'avons vu dans le chapitre 1, il existe quelques méthodes générales pour définir une telle fonction. Celle introduite dans le filtre particulaire auxiliaire [Pitt et Shephard, 1999] en est certainement la plus connue, de par son efficacité, sa simplicité algorithmique, et sa complexité, qui reste linéaire en fonction du nombre de particules (section 1.2.6).

Algorithme 3.1: Filtre particulaire pour l'intégration d'informations spatiales floues

Entrée : Approximation initiale $P_N(d\mathbf{s}_0) = \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{s}_0^{(n)}}(d\mathbf{s}_0)$ **Sortie** : Approximation P_N de la loi *a posteriori* de la trajectoire $\mathbb{P}(d\mathbf{s}_{0:T}|\mathbf{y}_{1:T}, \mathcal{Z})$ Notation : $\mathbf{s}_{0:t}^{(n)} \triangleq (\mathbf{x}_{0:t}^{(n)}, \mathbf{r}_{0:t}^{(n)}, \mathbf{c}_{0:t}^{(n)})$ début pour $t = 1, \ldots, T$ faire • pour n = 1, ..., N faire • Générer $\tilde{\mathbf{c}}_{t}^{(n)} \sim q(\mathbf{c}_{t} | \mathbf{c}_{a_{t}:t-1}^{(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z})$ (Eq. 3.18) • Générer $\tilde{\mathbf{r}}_{t}^{(n)} \sim \mathbb{P}(\mathbf{r}_{t} \in d\mathbf{r}_{t} | \tilde{\mathbf{c}}_{t}^{(n)}, \mathcal{Z})$ (Eq. 3.17) $\circ \text{ Générer } \mathbf{\tilde{x}}_{t}^{(n)} \sim q(\mathbf{x}_{t} | \mathbf{x}_{t-1}^{(n)}, \mathbf{\tilde{r}}_{t}^{(n)}) \\ \circ \text{ Poser } \mathbf{\tilde{s}}_{t}^{(n)} = (\mathbf{\tilde{x}}_{t}^{(n)}, \mathbf{\tilde{r}}_{t}^{(n)}, \mathbf{\tilde{c}}_{t}^{(n)}) \text{ et } \mathbf{\tilde{s}}_{0:t}^{(n)} = (\mathbf{s}_{0:t-1}^{(n)}, \mathbf{\tilde{s}}_{t}^{(n)})$ • Calculer les poids d'importance $\tilde{w}(\tilde{\mathbf{s}}_{t}^{(n)}) = w_{t-1}^{(n)} \frac{p(\mathbf{y}_{t} | \tilde{\mathbf{x}}_{t}^{(n)}, \tilde{\mathbf{r}}_{t}^{(n)}, \tilde{\mathbf{c}}_{t}^{(n)}) p(\tilde{\mathbf{x}}_{t}^{(n)} | \mathbf{x}_{t-1}^{(n)}, \tilde{\mathbf{r}}_{t}^{(n)}) \mathbb{P}(\tilde{\mathbf{c}}_{t}^{(n)} | \mathbf{c}_{t-1}^{(n)}, \mathcal{Z})}{q(\tilde{\mathbf{x}}_{t}^{(n)} | \mathbf{x}_{t}^{(n)}, \tilde{\mathbf{r}}_{t}^{(n)}) q(\tilde{\mathbf{c}}_{t}^{(n)} | \mathbf{c}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}$ • pour $n = 1, \ldots, N$ faire • Calculer les poids d'importance normalisés : $w_t^{*(n)} = \frac{\tilde{w}(\tilde{\mathbf{s}}_t^{(n)})}{\sum_{m=1}^N \tilde{w}(\tilde{\mathbf{s}}_t^{(m)})}$ • si $\hat{N}_{eff}(N) < N_{seuil}(N)$ alors • Rééchantillonner $\{\mathbf{s}_{0:t}^{(n)}, 1/N\}_{n=1}^N$ à partir du nuage $\{\tilde{\mathbf{s}}_{0:t}^{(n)}, w_t^{*(n)}\}_{n=1}^N$ selon l'algorithme du rééchantillonnage multinomial 1.9. sinon • Poser $\{\mathbf{s}_{0:t}^{(n)}, w_t^{(n)}\}_{n=1}^N = \{\tilde{\mathbf{s}}_{0:t}^{(n)}, w_t^{*(n)}\}_{n=1}^N$ • Approximation de la loi marginale au temps *t*, dite loi de filtrage : $\mathbb{P}(d\mathbf{s}_t|\mathbf{y}_{1:t}) \simeq P_N(d\mathbf{s}_t|\mathbf{y}_{1:t}) = \sum_{n=1}^N w_t^{(n)} \delta_{\mathbf{s}_t^{(n)}}(d\mathbf{s}_t)$ • Espérance de Monte-Carlo issue de la loi de filtrage : $\mathbb{E}_{p(\mathbf{s}_t|\mathbf{y}_{1:t})}[\phi(\mathbf{s}_t)] \simeq \sum_{t=1}^{N} w_t^{(n)} \phi(\mathbf{s}_t^{(n)})$

retourner $P_N(d\mathbf{s}_{0:T}|\mathbf{y}_{1:T}) = \sum_{n=1}^N w_T^{(n)} \delta_{\mathbf{s}_{0:T}^{(n)}}(d\mathbf{s}_{0:T})$

3.4.1 Modélisation de la fonction d'importance

Nous proposons dans cette section de définir une fonction d'importance dédiée au modèle développé. A la différence de celle utilisée dans le filtre particulaire auxiliaire, celle que nous proposons dépend de la dernière observation mais aussi de la trajectoire, c'est-à-dire de l'historique de l'état inconnu du système, représentée par les indices des m + 1 derniers ensembles flous estimés. L'idée que nous allons exploiter ici est de synthétiser la trajectoire des indices des ensembles flous, en tirant profit de leur modélisation qui est par nature discrète, et d'utiliser la dernière observation, afin de prédire les indices des ensembles flous qui seront les plus susceptibles d'être considérés au temps courant. Le principe de la fonction de proposition proposée est illustré sur la figure 3.3.



FIG. 3.3 – Illustration de la fonction de proposition. En bleu ciel, la trajectoire passée, qui contient par exemple l'orientation de l'objet, son échelle, sa forme, *etc.*, représentées sous forme d'informations spatiales, qui est ensuite utilisée pour estimer la position de l'objet bleu à l'instant courant. La fonction de proposition utilise la trajectoire passée ainsi que l'estimation de l'objet à l'instant courant afin de prédire l'état à l'instant suivant, caractérisé par un ensemble d'informations spatiales ainsi que d'une observation issue de l'image représentée en rouge.

Sous une hypothèse d'indépendance, la fonction d'importance s'écrit :

$$q(\mathbf{c}_{t}|\mathbf{c}_{a_{t}:t-1}^{(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}) = \prod_{r=1}^{R} q(\mathbf{c}_{t}^{r}|\mathbf{c}_{a_{t}:t-1}^{r,(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r})$$
$$= \prod_{r=1}^{R} \sum_{k=1}^{K^{r}} \delta_{k}(\mathbf{c}_{t}^{r}) q(\mathbf{c}_{t}^{r} = k|\mathbf{c}_{a_{t}:t-1}^{r,(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r})$$
(3.18)

où $a_t \triangleq t - 1 - m$ avec m une constante fixée qui permet de prendre en compte les m + 1 dernières réalisations de c (voir la section 4.2.2.1 pour des expériences à propos de l'influence du paramètre m).

Le dernier terme dans l'équation 3.18 est défini comme une loi de mélange de la fonction de transition définie dans l'équation 3.13 et une distribution qui extrait une information *a priori* à partir de la trajectoire de $c_{a_t:t-1}$ et de la nouvelle observation y_t :

$$q(\mathbf{c}_{t}^{r} = k | \mathbf{c}_{a_{t}:t-1}^{r,(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r}) = \gamma \mathbb{P}(\mathbf{c}_{t}^{r} = k | \mathbf{c}_{a_{t}:t-1}^{r,(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r}) + (1 - \gamma) \mathbb{P}(\mathbf{c}_{t}^{r} = k | \mathbf{c}_{t-1}^{r,(n)}, \mathcal{Z}^{r})$$
(3.19)

avec $\gamma \in [0, 1]$ un coefficient de pondération et $\mathbb{P}(\mathbf{c}_t^r = k | \mathbf{c}_{t-1}^{r,(n)}, \mathcal{Z}^r)$ la distribution de transition (équation 3.15). La valeur du paramètre γ sera discutée dans la section 4.2.2.2. Nous proposons maintenant de modéliser la distribution $\mathbb{P}(\mathbf{c}_t^r = k | \mathbf{c}_{a_t:t-1}^{r,(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_t, \mathcal{Z}^r)$. Pour ce faire, nous considérons les probabilités $\pi_t^k = \mathbb{P}(\mathbf{c}_t^r = k | \mathbf{c}_t^r)$

Pour ce faire, nous considérons les probabilités $\pi_r^k = \mathbb{P}(\mathbf{c}_t^r = k | \mathbf{c}_{a_t:t-1}^{r,(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{r,(n)}, \mathbf{y}_t, \mathcal{Z}^r)$ comme une réalisation du vecteur aléatoire $\Pi_r = (\Pi_r^1, \ldots, \Pi_r^{K^r})$ qui prend ses valeurs dans $[0, 1]^{K^r}$ tel que $\sum_{k=1}^{K^r} \Pi_r^k = 1$. Dans le but de simuler des échantillons selon la distribution $Pr(\mathbf{c}_t^r = k | \mathbf{c}_{a_t:t-1}^{r,(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{r,(n)}, \mathbf{y}_t, \mathcal{Z}^r)$, les probabilités $\pi_r = (\pi_r^1, \ldots, \pi_r^{K^r})$ sont générées selon la distribution d'importance décrite par l'équation suivante :

$$q(\pi_{r}|\mathbf{c}_{a_{t}:t-1}^{r,(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r}) \propto q(\mathbf{c}_{a_{t}:t-1}^{r,(n)} | \pi_{r}, \mathcal{Z}^{r}) q(\pi_{r}|\mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r}) = \mathcal{M}\left(\mathsf{nb}^{1}(\mathbf{c}_{a_{t}:t-1}^{r,(n)}), \dots, \mathsf{nb}^{K^{r}}(\mathbf{c}_{a_{t}:t-1}^{r,(n)})\right) \times \mathcal{D}\left(\pi_{r}; \left\{1 + \alpha_{k}^{r}q(\mathbf{c}_{t}^{r} = k|\mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r})\right\}_{k=1}^{K^{r}}\right) \propto \mathcal{D}\left(\pi_{r}; \left\{1 + \alpha_{k}^{r}q(\mathbf{c}_{t}^{r} = k|\mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r}) + \mathsf{nb}^{k}(\mathbf{c}_{a_{t}:t-1}^{r,(n)})\right\}_{k=1}^{K^{r}}\right)$$
(3.20)

où $\mathsf{nb}^j(l_{t_1:t_2}) = \sum_{i=t_1}^{t_2} \delta_j(l_i), (t_1, t_2) \in \mathbb{N}^2$. Les paramètres $\{\alpha_k^r\}_{k=1}^{K^r}$ sont les paramètres d'échelle de la loi de Dirichlet \mathcal{D} . Ils correspondent aux degrés d'importance des modes \mathbf{c}_t^r et gèrent la forme globale de la distribution *a priori* de Dirichlet. La distribution multinomiale \mathcal{M} utilise les m + 1 dernières réalisations de $\mathbf{c}_{a_t:t-1}^{r,(n)}$. La distribution de Dirichlet étant la loi *a priori* conjuguée de la distribution multinomiale, la loi *a posteriori* ainsi obtenue est une distribution de Dirichlet [Fink, 1997], qui intègre à la fois la trajectoire de c_t et les nouvelles probabilités *a priori* des modes. La proportionnalité obtenue dans l'équation 3.20 vient du fait que dans un souci de clarté, le dénominateur n'est pas représenté dans les deux premières lignes de l'équation. A notre connaissance, l'intégration d'une loi de Dirichlet dans un filtre particulaire n'a été effectué que dans [Hue et al., 2002; Caron et al., 2007]. Cependant, leur utilisation et la nôtre sont tout à fait différentes. Dans [Caron et al., 2007], elle sert à inférer les probabilités qu'un capteur soit dans un état donné, alors que dans [Hue et al., 2002], elle est utilisée dans un cadre multi-objets, et plus particulièrement dans un échantillonneur de Gibbs dans le but d'estimer les probabilités d'associations des données aux cibles. Dans notre approche, la loi de Dirichlet sert à synthétiser les informations issues de la trajectoire, ainsi qu'à intégrer la dernière mesure.

Rappelons maintenant l'expression de la densité de probabilité de la loi de Dirichlet $\mathcal{D}(u_1, \ldots, u_K; q_1, \ldots, q_K)$ de paramètres $q_1, \ldots, q_K > 0$, définissant un simplexe de K-1 degrés de liberté dont les paramètres u_1, \ldots, u_K satisfont $u_1, \ldots, u_{K-1} > 0$ et $\sum_{k=1}^{K-1} u_k < 1$, avec $u_K = 1 - u_1 - \ldots u_{K-1}$:

$$\mathcal{D}(u_1, \dots, u_K; q_1, \dots, q_K) = \frac{1}{\mathcal{B}(q_1, \dots, q_K)} \prod_{k=1}^K (u_k)^{q_k - 1}$$
$$= \frac{\Gamma(\sum_{k=1}^K q_k)}{\prod_{k=1}^K \Gamma(q_k)} \prod_{k=1}^K (u_k)^{q_k - 1}$$
(3.21)

avec \mathcal{B} la fonction bêta multinomiale et Γ la fonction gamma. La densité de probabilité de la loi de Dirichlet est illustrée sur la figure 3.4 en utilisant différents jeux de paramètres. De l'expression de la loi *a posteriori* définie dans l'équation 3.20, nous pouvons remarquer que les paramètres de la loi de Dirichlet, que nous notons q_1, \ldots, q_K , ont la particularité suivante : $q_1, \ldots, q_K \ge 1$. Cela signifie qu'un *a priori* uniforme est injecté dans la loi, tel

qu'il est possible de le constater dans l'équation 3.21 en prenant $q_1 = \ldots, q_K = 1$, ou sur la figure 3.4(b). Cette information uniforme sert à obtenir le comportement attendu de la loi *a posteriori* de Dirichlet, que nous n'aurions pas avec $q_1, \ldots, q_k < 1$ (figure 3.4(a)). En effet, la loi a posteriori de Dirichlet sert à inférer les paramètres de la loi discrète des indices des classes (équation 3.19). En récupérant plus d'informations, c'est-à-dire en considérant plus de réalisations dans la loi multinomiale (et donc en fixant une longueur md'historique plus grande), nous nous attendons à obtenir une surface de la densité de Dirichlet plus abrupte, et ainsi dessiner un « pic ». Le mode ainsi formé désigne la valeur (u_1, \ldots, u_K) dont la probabilité est maximale. Prenons pour exemple la densité de Dirichlet dont les paramètres valent q = (5, 5, 5). Nous pouvons imaginer le scénario dans lequel la loi multinomiale fournit 16 réalisations, 4 par classes, et dont les paramètres d'échelle sont égaux à $\{\alpha_k^r\}_{k=1}^{K^r} = 0$. Dans ce cas, nous obtenons bien le jeu de paramètres q = (5, 5, 5). A la lecture de la figure 3.4(c), le mode, qui donne la probabilité maximale, est alors nettement atteint pour la valeur u = (1/3, 1/3, 1/3). Autrement dit, nous obtenons une probabilité maximale de générer une loi des indices des classes uniforme. Et bien évidemment, plus les valeurs des paramètres sont grandes, plus la forme de la densité de Dirichlet est « piquée » (figures 3.4(c) et 3.4(d)), ce qui correspond bien à l'effet souhaité. Enfin, les probabilités $(\pi_r^1, \ldots, \pi_r^{K^r})$ sont obtenues en simulant selon la distribution *a pos*teriori de Dirichlet (équation 3.20), ce qui rend la simulation selon l'équation 3.19 directe. Ce sont toutes ces considérations qui nous ont conduits à proposer le modèle exprimé dans l'équation 3.20.



FIG. 3.4 – Illustrations de la densité de probabilité de Dirichlet en dimension 3, avec $u_3 = 1 - u_1 - u_2$, de paramètres (a) q = (0.3, 0.3, 0.3), (b) q = (1, 1, 1), (c) q = (5, 5, 5) et (d) q = (3, 7, 1).

Il reste maintenant à déterminer la distribution $q(\mathbf{c}_t^r = k | \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_t, \mathcal{Z}^r)$ présente

dans l'équation 3.20. Celle-ci peut s'écrire comme une loi de mélange :

$$q(\mathbf{c}_{t}^{r} = k | \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r}) = \beta^{1} \mathbb{P}(\mathbf{c}_{t}^{r} = k | \mathbf{r}_{t-1}^{r,(n)}, \mathcal{Z}^{r}) + \beta^{2} \mathbb{P}(\mathbf{c}_{t}^{r} = k | \mathbf{x}_{t-1}^{(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r}) + \beta^{3} \mathbb{P}(\mathbf{c}_{t}^{r} = k | \mathcal{Z}^{r})$$
(3.22)

où $\sum_{k=1}^{3} \beta^{k} = 1$, et dont l'influence des valeurs sera discutée dans la section 4.2.2.2. $\mathbb{P}(\mathbf{c}_{t}^{r} = k | \mathcal{Z}^{r})$ est la distribution *a priori* définie par l'équation 3.13, et $\mathbb{P}(\mathbf{c}_{t}^{r} = k | \mathbf{r}_{t-1}^{r,(n)}, \mathcal{Z}^{r})$ est la probabilité *a posteriori* de considérer au temps *t* le $k^{\text{ième}}$ ensemble flou représentant une valeur du $r^{\text{ième}}$ concept flou conditionné par la réalisation passée du paramètre $\mathbf{r}_{t-1}^{r,(n)}$. Le calcul de la distribution $\mathbb{P}(\mathbf{c}_{t}^{r} = k | \mathbf{x}_{t-1}^{(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r})$ est déterminant car celleci est conditionnée par la dernière observation \mathbf{y}_{t} . Ainsi, en pratique, il est fréquent de considérer un jeu de paramètres tel que $\beta^{2} \gg \beta^{1}$ et $\beta^{2} \gg \beta^{3}$. Par le théorème de Bayes, nous obtenons :

$$\mathbb{P}(\mathbf{c}_{t}^{r} = k | \mathbf{x}_{t-1}^{(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r}) = \frac{p(\mathbf{y}_{t} | \mathbf{c}_{t}^{r} = k, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathcal{Z}^{r}) p(\mathbf{c}_{t}^{r} = k, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathcal{Z}^{r})}{\sum_{j=1}^{K^{r}} p(\mathbf{y}_{t} | \mathbf{c}_{t}^{r} = j, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathcal{Z}^{r}) p(\mathbf{c}_{t}^{r} = j, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathcal{Z}^{r})}$$
(3.23)

Malheureusement, les termes de droite présents au numérateur et au dénominateur de l'équation 3.23 ne peuvent être estimés de manière analytique, en raison de la présence de $\mathbf{x}_{t-1}^{(n)}$. Une approximation de ce terme serait bien sûre possible, mais nous avons plutôt opté pour une simplification radicale de l'expression 3.23, en ne considérant pas les termes $p(\mathbf{c}_t^r, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathcal{Z}^r)$. Cela permet de bien mettre en évidence le terme de vraisemblance $p(\mathbf{y}_t | \mathbf{c}_t^r = k, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathcal{Z}^r)$, tout en proposant un calcul analytique simple de la distribution $\mathbb{P}(\mathbf{c}_t^r = k | \mathbf{x}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathbf{y}_t, \mathcal{Z}^r)$:

$$\mathbb{P}(\mathbf{c}_{t}^{r} = k | \mathbf{x}_{t-1}^{(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{y}_{t}, \mathcal{Z}^{r}) \approx \frac{p(\mathbf{y}_{t} | \mathbf{c}_{t}^{r} = k, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathcal{Z}^{r})}{\sum_{j=1}^{K^{r}} p(\mathbf{y}_{t} | \mathbf{c}_{t}^{r} = j, \mathbf{r}_{t-1}^{(n)}, \mathbf{x}_{t-1}^{(n)}, \mathcal{Z}^{r})} \\
= \frac{p(\mathbf{y}_{t} | g_{t}(\mathbf{x}_{t-1}^{(n)}, \tilde{\mathbf{r}}_{t|t-1}^{r,k,(n)}, \mathbf{v}_{t}), \mathcal{Z}^{r})}{\sum_{j=1}^{K^{r}} p(\mathbf{y}_{t} | g_{t}(\mathbf{x}_{t-1}^{(n)}, \tilde{\mathbf{r}}_{t|t-1}^{r,k,(n)}, \mathbf{v}_{t}), \mathcal{Z}^{r})} \qquad (3.24)$$

où

$$\tilde{\mathbf{r}}_{t|t-1}^{r_k,(n)} = \left(\mathbf{r}_{t-1}^{1,(n)}, \dots, \mathbf{r}_{t-1}^{r-1,(n)}, \bar{\mathbf{r}}_t^{r_k}, \mathbf{r}_{t-1}^{r+1,(n)}, \dots, \mathbf{r}_{t-1}^{R,(n)}\right)$$
(3.25)

est une valeur candidate du paramètre $\mathbf{r}_{t|t-1}$, et $\mathbf{\bar{r}}_t^{r_k}$ est une caractéristique de la fonction d'appartenance $\mu_k^r(\mathbf{r})$, par exemple sa valeur moyenne. La probabilité $\mathbb{P}(\mathbf{c}_t^r = k|\mathbf{x}_{t-1}^{(n)}, \mathbf{r}_{t-1}^{(n)}, \mathbf{y}_t, \mathcal{Z}^r)$ est donc proportionnelle à la vraisemblance issue de la prédiction à t de \mathbf{x}_{t-1} en utilisant comme r-ième paramètre la valeur $\mathbf{\bar{r}}_t^{r_k}$. Notons que l'approximation 3.25 semble raisonnable dans la mesure où la probabilité d'une classe de rester dans le même état est maximale (selon l'équation 3.15).

Une stratégie plus complexe peut être employée pour proposer un candidat $\tilde{\mathbf{r}}_{t|t-1}^{r_k,(n)}$ plus pertinent. En utilisant la formule des probabilités totales, elle consiste à intégrer les

valeurs candidates de tous les paramètres, sauf de celui qui nous intéresse, c'est-à-dire le *r*-ième. Le candidat est généré selon la distribution suivante :

$$\tilde{\mathbf{r}}_{t}^{r_{k},(n)} \sim \frac{1}{U} \sum_{j_{1}=1}^{K^{1}} \dots \sum_{j_{r-1}=1}^{K^{r-1}} \sum_{j_{r+1}=1}^{K^{r+1}} \dots \sum_{j_{R}=1}^{K^{R}} \prod_{\forall h \neq r} \delta_{\tilde{\mathbf{r}}_{t}^{h_{j_{h}}}} \left(\tilde{\mathbf{r}}_{t}^{h,(n)} \right) p(\mathbf{y}_{t}|g_{t}(\mathbf{x}_{t-1}^{(n)}, \tilde{\mathbf{r}}_{t}^{r_{k},(n)}, \mathbf{v}_{t}), \mathcal{Z}^{r})$$
(3.26)

avec U le facteur de normalisation tel que

$$U = \sum_{j_1=1}^{K^1} \dots \sum_{j_{r-1}=1}^{K^{r-1}} \sum_{j_{r+1}=1}^{K^{r+1}} \dots \sum_{j_R=1}^{K^R} p(\mathbf{y}_t | g_t(\mathbf{x}_{t-1}^{(n)}, \bar{\mathbf{r}}_t^{j*}, \mathbf{v}_t), \mathcal{Z}^r)$$

et où

$$\tilde{\mathbf{r}}_{t}^{r_{k},(n)} = \left(\tilde{\mathbf{r}}_{t}^{1,(n)}, \dots, \tilde{\mathbf{r}}_{t}^{r-1,(n)}, \bar{\mathbf{r}}_{t}^{r_{k}}, \tilde{\mathbf{r}}_{t}^{r+1,(n)}, \dots, \tilde{\mathbf{r}}_{t}^{R,(n)}\right)$$
(3.27)

$$\bar{\mathbf{r}}_{t}^{j*} = (\bar{\mathbf{r}}_{t}^{1_{j_{1}}}, \dots, \bar{\mathbf{r}}_{t}^{r-1_{j_{r-1}}}, \bar{\mathbf{r}}_{t}^{r_{k}}, \bar{\mathbf{r}}_{t}^{r+1_{j_{r+1}}}, \dots, \bar{\mathbf{r}}_{t}^{R_{j_{R}}})$$
(3.28)

Dans l'expression 3.26, le candidat est donc généré proportionnellement à la vraisemblance associée aux paramètres $\tilde{\mathbf{r}}_{t}^{r_{k},(n)}$, et correspond donc à une configuration des R-1paramètres, conditionnellement au r-ième fixé à la valeur $\bar{\mathbf{r}}_{t}^{r_{k}}$. Comme dans l'équation 3.25, les paramètres de référence correspondent à des caractéristiques des fonctions d'appartenance concernées, telles que la moyenne. La distribution ainsi définie prend en compte l'ensemble des valeurs des paramètres possibles, et n'a plus recourt à l'approximation consistant à considérer les réalisations à t - 1 des paramètres (équation 3.25). Cette stratégie, bien que clairement plus robuste que celle définie par l'équation 3.24, n'a pas été considérée dans nos tests. En effet, elle se trouve être très coûteuse en termes de temps de calculs, car sa complexité est exponentielle en fonction du nombre de classes par paramètre. Elle n'est donc utilisable en pratique que lorsque la cardinalité de \mathbf{c}_{t} est raisonnable, et qu'un petit nombre de valeurs est considéré par classe.

3.4.2 Complexité et coût de calcul

Un dernier point important concerne l'étude du coût de calcul de la fonction d'importance proposée. D'après l'équation 3.18, cette quantité dépend du nombre de paramètres, ainsi que du nombre de classes modélisées par paramètre, soit $\sum_{r=1}^{R} K^r$ opérations par particule. Cela peut être coûteux en particulier pour simuler selon la distribution 3.20, qui fait appel à l'évaluation de la vraisemblance utilisée dans l'équation 3.24. Or, en traitement d'images, celle-ci est généralement gourmande en temps de calcul, à tel point que les performances en temps du filtre particulaire sont directement liées au nombre de fois où la vraisemblance est evaluée. Heureusement, il est possible de diminuer ce nombre en calculant la distribution 3.20 uniquement pour des particules distinctes, tirant ainsi profit du rééchantillonnage effectué au temps précédent, lorsque celui-ci a eu lieu. Cette astuce algorithmique ne doit pas être ignorée puisqu'elle représente en moyenne, et approximativement, une économie du nombre de calculs de la vraisemblance égale à $\hat{N}_{\text{eff}} \sum_{r=1}^{R} K^r$ (section 1.2.2). Une autre solution, plus *ad hoc* mais tout aussi efficace, consiste à considérer dans l'équation 3.24 une version simplifiée de la vraisemblance, lorsque cela est possible. Cela sera par exemple le cas dans l'expérience réalisée dans la section 4.3.2.2.

3.4.3 Conclusion

Pour résumer, la modélisation discrète de la fonction d'importance permet de facilement synthétiser la trajectoire de l'état, et ainsi de garder des hypothèses dans le temps. Cela permet d'obtenir, et d'utiliser, une information globale. Ainsi, la fonction d'importance proposée a un comportement adaptatif, ce qui permet de gérer des changements brusques dans la dynamique, comme les valeurs aberrantes, tout en utilisant une information de trajectoire pour passer outre les ambiguïtés locales, comme nous le verrons dans le chapitre 4. Par exemple, de par l'utilisation de la trajectoire des indices des fonctions floues et en considérant un modèle de vélocité constant, en présence d'un mouvement erratique, la fonction d'importance tend à produire une exploration uniforme de l'espace d'état des paramètres, alors que dans les cas d'une séquence très linéaire, une région spécifique sera plus vraisemblablement considérée. Bien sûr, cela reste conditionné notamment par la dernière observation, et donc par les vraisemblances obtenues par les différentes hypothèses des paramètres. Cela est illustré sur la figure 3.5.



FIG. 3.5 – Comportement de la fonction d'importance avec \mathbf{x}_t la position de l'objet à l'instant t et dont la dynamique est caractérisée par une information de vélocité floue (dont les valeurs pourraient être *Nord-Ouest* à *Sud-Est*). a) En présence d'une trajectoire erratique, la fonction n'infère aucune hypothèse, la loi de simulation tend à être uniforme. b) En prenant en compte toute la trajectoire, la fonction attribue la même probabilité aux deux hypothèses. c) Bien que peu probable du point de vue de la trajectoire, l'hypothèse au *Sud-Ouest* n'est pas écartée, surtout si elle atteint une observation \mathbf{y}_t . Cela est possible parce que la fonction d'importance prend à la fois en compte la trajectoire et la vraisemblance.

3.5 Modélisation d'informations spatiales floues en pratique

3.5.1 Modélisation des relations spatiales

En considérant les relations spatiales décrites dans la section 2.2, avec la fonction d'appartenance de l'ensemble des possibles du paramètre r tel que $\mu_{Z}^{r} = \sum_{k=1}^{K^{r}} \mu_{k}^{r}$, les expressions des équations 3.11, 3.13, 3.14 et 3.15 se simplifient et nous obtenons :

$$\mathbb{P}(\mathbf{r}^{r} \in d\mathbf{r}^{r} | \mathcal{Z}^{r}) = \frac{d\mathbf{r}^{r} \mu_{\mathcal{Z}}^{r}(\mathbf{r}^{r})}{\int_{\mathcal{R}^{r}} \mu_{\mathcal{Z}}^{r}(\mathbf{r}') d\mathbf{r}'} = \frac{d\mathbf{r}^{r}}{|\mathcal{R}^{r}|}$$
(3.29)

$$\mathbb{P}(\mathbf{c}^{r} = k | \mathbf{r}^{r}, \mathcal{Z}^{r}) = \frac{\mu_{k}^{r}(\mathbf{r}^{r})}{\sum_{h=1}^{K} \mu_{h}^{r}(\mathbf{r}^{r})} = \mu_{k}^{r}(\mathbf{r}^{r})$$
(3.30)

Cette même propriété de sommation a été utilisée pour obtenir l'expression :

$$\mathbb{P}(\mathbf{c}^{r} = k | \mathcal{Z}^{r}) = \int_{\mathcal{R}^{r}} \frac{\mu_{k}^{r}(\mathbf{r})}{\sum_{h=1}^{K} \mu_{h}^{r}(\mathbf{r})} \frac{\mu_{\mathcal{Z}}^{r}(\mathbf{r})}{\int_{\mathcal{R}^{r}} \mu_{\mathcal{Z}}^{r}(\mathbf{r}') d\mathbf{r}'} d\mathbf{r}$$

$$= \frac{\int_{\mathcal{R}^{r}} \mu_{k}^{r}(\mathbf{r}) d\mathbf{r}}{\int_{\mathcal{R}^{r}} \mu_{\mathcal{Z}}^{r}(\mathbf{r}') d\mathbf{r}'}$$

$$= \frac{1}{K^{r}}$$
(3.31)

Le passage à la dernière ligne se fait en considérant la propriété définie dans l'équation 2.5, qui contraint les intégrales sur \mathcal{R}^r des fonctions d'appartenance $\{\mu_k^r\}_{k=1}^{K^r}$ à être identiques, ce qui permet également de simplifier l'expression de l'équation suivante :

$$\mathbb{P}(\mathbf{r}^{r} \in d\mathbf{r}^{r} | \mathbf{c}^{r} = k, \mathcal{Z}^{r}) = \frac{d\mathbf{r}^{r} \mu_{k}^{r}(\mathbf{r}^{r}) \mu_{\mathcal{Z}}^{r}(\mathbf{r}^{r})}{\sum_{h=1}^{K} \mu_{h}^{r}(\mathbf{r}^{r})} \Big/ \int_{\mathcal{R}^{r}} \frac{\mu_{k}^{r}(\mathbf{r}') \mu_{\mathcal{Z}}^{r}(\mathbf{r}')}{\sum_{h=1}^{K} \mu_{h}^{r}(\mathbf{r}')} d\mathbf{r}' \\
= \frac{d\mathbf{r}^{r} \mu_{k}^{r}(\mathbf{r}^{r})}{\int_{\mathcal{R}^{r}} \mu_{k}^{r}(\mathbf{r}') d\mathbf{r}'} \\
= \frac{d\mathbf{r}^{r} \mu_{k}^{r}(\mathbf{r}^{r})}{|\mathcal{R}^{r}|/K^{r}} \tag{3.32}$$

Il est également possible de définir des relations spatiales par apprentissage. Par exemple, en utilisant une stratégie de fusion intra-classes disjonctive et une stratégie inter-classes normalisatrice, nous pourrions obtenir les mêmes expressions que celles décrites dans les équations 3.29 et 3.30.

Comme nous l'avons vu dans la section 2.2, il est possible de facilement former une partition floue décrivant des relations spatiales en considérant des fonctions d'appartenance $\{\mu_k^r\}_{k=1}^{K^r}$ de formes trapézoïdales. Ainsi, générer un échantillon selon l'équation 3.32 s'effectue en utilisant une méthode d'inversion de la fonction de répartition (section 1.1.2.1). Le procédé de simulation selon une fonction trapézoïdale est présenté dans l'algorithme 3.2.

3.5.2 Modélisation d'informations spatiales générales

Lorsque les contraintes énoncées dans le paragraphe précédent sont trop fortes, ou lorsque nous souhaitons modéliser une information spatiale plus générale, il convient simplement de définir un ensemble flou Z^r des valeurs possibles qui soit pertinent, au sens du modèle que nous souhaitons définir. Une liste non exhaustive de ces stratégies a été étudiée dans la section 2.1.4.

Dans un cas général, c'est-à-dire lorsque les ensembles flous sont appris et caractérisés par des formes complexes, il est souvent nécessaire d'approcher les distributions des paramètres, ce qui est le sujet de la prochaine section. Algorithme 3.2: Algorithme de simulation selon une fonction floue de forme trapézoïdale

3.6 Approximation

Dans certains cas (stratégie probabiliste, fonction d'appartenance se décomposant sur chacune des dimensions de \mathcal{R}^r , *etc.*), le calcul et/ou la simulation selon les équations 3.13, 3.11, 3.14 et 3.15 peut se faire de manière analytique (ou directe). Dans les autres cas, il est nécessaire de recourir à une méthode d'approximation. Nous proposons ici de réaliser l'approximation de la distribution *a priori* $\mathbb{P}(\mathbf{r}^r \in d\mathbf{r}^r | \mathcal{Z}^r)$ via l'algorithme de Metropolis-Hasting indépendant, bien qu'il soit également possible d'utiliser un algorithme d'acceptation-rejet (en modifiant quelque peu la fonction de proposition de simulation), étudiés dans le chapitre 1. Cependant, l'algorithme d'acceptation-rejet nécessite de connaître la valeur $k \in \mathbb{R}$ minimale telle que $\forall \mathbf{r}^r \in \mathcal{R}^r, kG(\mathbf{r}^r) \ge F(\mathbf{r}^r)$, avec *G* la fonction d'importance et *F* la loi cible, ce qui peut compliquer la mise en œuvre du processus d'approximation. Nous renvoyons à [Liu, 1996] pour une comparaison en termes de convergence de ces deux algorithmes. Nous posons :

$$F(\mathbf{r}^r) = \mu_{\mathcal{Z}}(\mathbf{r}^r) \quad \text{et} \quad G(\mathbf{r}^r) = \sum_{k=1}^{K^r} \sum_{m=1}^{M^k} P_{\mathbf{o}_m^k}(\mathbf{r}^r)$$

Les fonctions F et G correspondent aux versions non normalisées de fonctions de densités de probabilités, la première correspondant à $\mathbb{P}(\mathbf{r}^r \in d\mathbf{r}^r | \mathcal{Z}^r)$ alors que G aurait pu être obtenue par un opérateur de fusion intra-classes additif, sans stratégie inter-classes ni contrainte. Ainsi, le support de G inclut celui de F, et a une forme qui permet de facilement générer des échantillons. Le processus d'approximation peut être réalisé par l'algorithme de Metropolis-Hasting présenté dans la section 1.1.3.2, où $G(\mathbf{r}^r)$ est la fonction de proposition, et $F(\mathbf{r}^r)$ la densité cible non normalisée de $\mathbb{P}(\mathbf{r}^r \in d\mathbf{r}^r | \mathcal{Z}^r)$. Soit $\{\mathbf{r}^{r,(l)}\}_{l=1}^L$

$$\mathbb{P}(\mathbf{r}^r \in d\mathbf{r}^r | \mathcal{Z}^r) \simeq \frac{1}{L} \sum_{l=1}^L \delta_{\mathbf{r}^{r,(l)}}(d\mathbf{r}^r)$$
(3.33)

Nous en déduisons directement l'expression des équations 3.13 et 3.14 :

l'approximation par un échantillon de taille L de la distribution a priori :

$$\mathbb{P}(\mathbf{c}^{r} = k | \mathcal{Z}^{r}) = \int_{\mathcal{R}^{r}} \mathbb{P}(\mathbf{c}^{r} = k | \mathbf{r}, \mathcal{Z}^{r}) \mathbb{P}(\mathbf{r}^{r} | \mathcal{Z}) d\mathbf{r}$$

$$\simeq \int_{\mathcal{R}^{r}} \frac{\mu_{k}^{r}(\mathbf{r})}{\sum_{h=1}^{K} \mu_{h}^{r}(\mathbf{r})} \frac{1}{L} \sum_{l=1}^{L} \delta_{\mathbf{r}^{r,(l)}}(\mathbf{r}) d\mathbf{r}$$

$$= \frac{1}{L} \sum_{l=1}^{L} \frac{\mu_{k}^{r}(\mathbf{r}^{r,(l)})}{\sum_{h=1}^{K} \mu_{h}^{r}(\mathbf{r}^{r,(l)})}$$
(3.34)

$$\mathbb{P}(\mathbf{r}^{r} \in d\mathbf{r}^{r} | \mathbf{c}^{r} = k, \mathcal{Z}^{r}) = \frac{\mathbb{P}(\mathbf{c}^{r} = k | \mathbf{r}^{r}, \mathcal{Z}^{r}) \mathbb{P}(\mathbf{r}^{r} \in d\mathbf{r}^{r} | \mathcal{Z}^{r})}{\mathbb{P}(\mathbf{c}^{r} = k | \mathcal{Z}^{r})} \\
\simeq \frac{1}{\mathbb{P}(\mathbf{c}^{r} = k | \mathcal{Z}^{r})} \frac{d\mathbf{r}^{r} \mu_{k}(\mathbf{r}^{r})}{\sum_{h=1}^{K} \mu_{h}(\mathbf{r}^{r})} \frac{1}{L} \sum_{l=1}^{L} \delta_{\mathbf{r}^{r,(l)}}(d\mathbf{r}^{r}) \\
= \frac{1}{\mathbb{P}(\mathbf{c}^{r} = k | \mathcal{Z}^{r})} \frac{1}{L} \sum_{l=1}^{L} \frac{\mu_{k}^{r}(\mathbf{r}^{r}) \delta_{\mathbf{r}^{r,(l)}}(d\mathbf{r}^{r})}{\sum_{h=1}^{K} \mu_{h}^{r}(\mathbf{r}^{r})} \tag{3.35}$$

alors que l'équation 3.12 reste inchangée. Ainsi, grâce à l'approximation réalisée dans l'équation 3.33, nous obtenons les formes analytiques des équations 3.13 et 3.14, ce qui permet de facilement les manipuler en pratique. Dans l'exemple des formes de bouches (section 2.3.2), nous obtenons ainsi un échantillon de L éléments de bouches, issues de la construction des ensembles flous via la base de données. Notons que cette étape d'approximation nécessite d'être réalisée une seule fois, lors de l'apprentissage des fonctions floues.

La forme de la fonction de proposition G utilisée dans le processus d'approximation est simple, ce qui peut ralentir de manière conséquente le processus d'approximation, et cela spécialement lorsque les fonctions d'appartenance sont fortement contraintes (fonction floue C de l'équation 2.2), ce qui peut leur donner des formes très complexes. Dû à l'échantillonnage préférentiel par nature utilisé dans l'algorithme de Metropolis-Hasting, le taux d'acceptation d'un nouveau candidat peut décroître de manière exponentielle en fonction de la dimension de \mathbf{r}^r , et cet effet est bien sûr clairement identifié lorsque la fonction de proposition est naïve. Pour résoudre ce problème, il est possible de relâcher légèrement la contrainte C des fonctions d'appartenance, afin d'améliorer le taux d'acceptation. Une autre solution, bien plus intéressante, serait d'améliorer le processus d'approximation, par exemple en construisant une fonction de proposition G plus proche de F. Cela peut être mis en perspective de notre travail.

3.7 Conclusion

Nous avons proposé dans ce chapitre une introduction originale d'informations spatiales floues dans le filtre particulaire, en considérant les ensembles flous comme des événements flous sur lesquels nous évaluons des probabilités. La famille de fonctions floues définit l'espace des paramètres, et forme un modèle dynamique à saut, dans lequel chaque état désigne un ensemble flou de la famille considérée. L'espace des paramètres est modélisé de manière probabiliste par une loi uniforme, rendant ainsi la manipulation des paramètres dynamiques entièrement gouvernée par l'information floue introduite.

Lorsque la structure des ensembles flous est complexe, comme c'est le cas par exemple avec les formes floues (section 2.3), il est nécessaire d'approcher la loi *a priori* des paramètres. Les autres lois se déduisent simplement de la première. La phase d'approximation est effectuée hors ligne, grâce à un algorithme de Metropolis-Hasting indépendant.

Nous avons également proposé une fonction d'importance qui dépend de la trajectoire des ensembles flous et des nouvelles observations. La combinaison de la trajectoire et de la nouvelle observation permet d'effectuer un compromis entre les informations apportées par la nouvelle donnée et les réalisations passées. Par exemple, dans le cas d'une nouvelle observation qui s'agirait d'une fausse alarme, utiliser la trajectoire permet de lisser la prédiction vers des zones cohérentes vis-à-vis de celle-ci, et ainsi atténuer le biais introduit par la fausse alarme. Toutefois, nous ne pouvons jamais savoir si une observation éloignée correspond réellement à une fausse alarme, et l'utilisation de la vraisemblance dans le processus de simulation permet de prendre en compte l'observation quoi qu'il arrive, ce qui permet de trouver un compromis entre utilisation de la trajectoire et nouvelle observation. Ce sont les procédures de rééchantillonnage futures qui détermineront la véracité des observations antérieures, en dupliquant les particules pertinentes.

Dans le chapitre suivant, nous mettons en pratique le filtre présenté ici, en modélisant des relations spatiales, des informations de formes floues, et en étudiant l'influence des divers paramètres introduits. Les relations spatiales floues définissent un modèle de suivi souple, c'est pourquoi nous comparons ce modèle dynamique robuste à d'autres approches faisant état de l'art dans ce domaine.

Chapitre 4

Expérimentations sur l'intégration d'informations spatiales floues dans le filtre particulaire

Sommaire

4.1	Prései	ntation de	es séquences de tests	
	4.1.1	ces synthétiques		
		4.1.1.1	Modèles dynamiques	
		4.1.1.2	Modèle de vraisemblance	
	4.1.2	Séquenc	ce de l'hélicoptère	
		4.1.2.1	Modèles dynamiques	
		4.1.2.2	Modèle de vraisemblance	
	4.1.3	Séquenc	ce de bouche	
		4.1.3.1	Modèle dynamique	
		4.1.3.2	Modèle de vraisemblance	
4.2	Influe	nce des p	paramètres	
	4.2.1 Modèle flou			
		4.2.1.1	Granularité de la famille de fonctions d'appartenance . 108	
		4.2.1.2	Formes des fonctions d'appartenance et paramètre ϵ 108	
		4.2.1.3	Longueur du support	
	4.2.2	n d'importance		
		4.2.2.1	Longueur de la trajectoire : paramètre m	
		4.2.2.2	Discussion à propos des autres paramètres	
4.3	Les re	lations sp	patiales floues en pratique	
	4.3.1	Adapta	bilité du modèle	
		4.3.1.1	Scénario synthétique 1	
		4.3.1.2	Scénario synthétique 2	
		4.3.1.3	Séquence d'hélicoptère	
	4.3.2	Temps o	de calcul et robustesse des modèles proposés	
		4.3.2.1	Scénario synthétique 3	
		4.3.2.2	Séquence d'hélicoptère	
4.4	L'info	L'information de formes floues en pratique		
4.5	Conclusion			

Les relations spatiales floues, de par leurs faibles hypothèses sur l'espace des paramètres, permettent de suivre un objet dont le modèle dynamique est mal connu, et potentiellement complexe. Dans ce contexte, nous avons choisi de comparer notre approche au filtre particulaire à modèles multiples (MMPF) et au filtre particulaire auxiliaire à modèles multiples (AUX-MMPF), qui sont des approches de référence dans la communauté de suivi par filtre particulaire avec sauts de modèles [Arulampalam et al., 2004]. En effet, comme nous l'avons vu dans la section 1.3.2, le MMPF et l'AUX-MMPF effectuent des sauts de modèles dynamiques afin de suivre un objet dont la dynamique est changeante, ce qui peut également être intéressant dans les cas où la dynamique est mal connue ou trop complexe pour être représentée par un seul modèle. Dans la mesure où nous étudierons le comportement de notre approche face à des situations dans lesquelles la dynamique est instable, tantôt linéaire, tantôt erratique, les modèles MMPF et AUX-MMPF semblent être pertinents comme base de comparaison. Nous présenterons également les résultats du filtre SIR et du filtre particulaire auxiliaire (AUX-PF), qui sont très largement utilisés dans la communauté, grâce à leur simplicité de mise en œuvre, et leur faculté d'adaptation à tous types de problématiques.

Pour illustrer à la fois la pertinence de la modélisation floue et la fonction d'importance proposée pour suivre les objets dans les différents scénarios, nous comparons les résultats obtenus avec notre modèle sans la fonction d'importance (que nous noterons FS-PF, pour *Fuzzy Spatial Particle Filter*), et avec la fonction d'importance (que nous noterons PFS-PF, pour *Proposal Fuzzy Spatial Particle Filter*), Ainsi, pour le FS-PF, les indices des ensembles flous seront générés selon l'équation 3.16, et n'utilisent donc pas la dernière observation durant l'étape de simulation.

Après avoir présenté les séquences de tests utilisées dans ce chapitre ainsi que les modèles de suivi considérés (section 4.1), nous étudierons l'influence des paramètres dans les modèles proposés (section 4.2). La section 4.3 traite du comportement des approches floues utilisant des relations spatiales. Enfin, nous présenterons dans la section 4.4 une application de suivi de bouche, en utilisant le modèle flou de forme proposé dans la section 2.3.

D'autres résultats que ceux présentés dans ce chapitre sont également disponibles dans les annexes B.2 et B.3.

4.1 Présentation des séquences de tests

Nous présentons dans cette section les séquences de tests ainsi que les modèles qui seront utilisés tout au long de ce chapitre. Un récapitulatif des séquences définies dans cette section est proposé dans le tableau 4.1.

Séquences	Définitions	Figures
Synthétiques 1,2,3	4.1.1	4.1(a), (b) et (c)
Hélicoptère	4.1.2	4.3 (a), (b) et (c)
Bouche	4.1.3	4.6(a), (b) et (c)

TAB. 4.1 – Récapitulatif des séquences définies dans la section 4.1.

4.1.1 Séquences synthétiques

Dans ce chapitre, nous utiliserons trois scénarios synthétiques présentés sur la figure 4.1. Les scénarios ont été générés à la main, dans le but de recréer des conditions classiques de suivi, dans lesquelles la dynamique de l'objet est mal connue, ce qui en pratique est très souvent le cas. Les deux premiers scénarios 4.1(a) et 4.1(b) présentent respectivement des caractéristiques de linéarité et de non linéarité volontairement exarcerbées, afin de facilement mettre en évidence le comportement des approches face à des situations extrêmes. La séquence 4.1(c) mêle des linéarités locales et des changements brutaux de la trajectoire, ce qui permet d'étudier le comportement des approches dans un contexte classique de suivi d'objets.



FIG. 4.1 – (a) Scénario synthéthique 1 : séquence linéaire, (b) scénario synthéthique 2 : séquence avec une dynamique erratique, (c) scénario synthéthique 3 : séquence de suivi classique.

4.1.1.1 Modèles dynamiques

Dans les séquences synthétiques, nous proposons d'estimer conjointement la position et la dynamique $\mathbf{x}_t = [\mathbf{x}_t \ \mathbf{y}_t]^T$ et $\mathbf{r}_t = [\dot{\mathbf{x}}_t \ \dot{\mathbf{y}}_t]^T$.

Modèles flous FS-PF, PFS-PF. Nous considérons le système dynamique de vélocité constante suivant :

$$\mathbf{x}_{t} = f_{t}(\mathbf{x}_{t-1}, \mathbf{r}_{t}, \mathbf{v}_{t})$$

$$= \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \end{bmatrix} \begin{bmatrix} \mathbf{x}_{t-1} \\ \mathbf{r}_{t} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^{2}}{2} & 0 \\ 0 & \frac{\Delta t^{2}}{2} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{t}^{1} \\ \mathbf{v}_{t}^{2} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{x}_{t-1} + \Delta t \, \dot{\mathbf{x}}_{t} + \frac{\Delta t^{2}}{2} \, \mathbf{v}_{t}^{1} \\ \mathbf{y}_{t-1} + \Delta t \, \dot{\mathbf{y}}_{t} + \frac{\Delta t^{2}}{2} \, \mathbf{v}_{t}^{2} \end{bmatrix}$$
(4.1)

où $[\mathbf{v}_t^1 \ \mathbf{v}_t^2]^T \sim \mathcal{N}(\mathbf{0}_{2 \times 1}, 4^2 \mathbf{I}_{2 \times 2})$, avec $\mathbf{0}_{2 \times 1} = [0 \ 0]^T$ et $\mathbf{I}_{2 \times 2}$ la matrice identité 2×2 .

Le paramètre de vélocité est estimé par notre formalisme flou. Nous considérons neuf valeurs linguistiques pour le couple $[\dot{x}_t \ \dot{y}_t]^T$, de *NordOuest* à *SudEst* modélisant la position relative d'un objet au temps t par rapport à sa position à t - 1. Ces paramètres sont modélisés d'une manière souple en définissant leurs ensembles flous :

 $\mathbf{c}_t^1 \in \{NO, N, NE, O, Centre, E, SO, S, SE\}$

dont les fonctions d'appartenance ont été définies dans la section 2.2.1 et sont illustrées sur la figure 4.2.



FIG. 4.2 – Partition uniforme floue bidimensionnelle du paramètre de vélocité. Les valeurs vont de la vélocité (a) *Nord-Ouest* à (i) *Sud-Est*. Un niveau de gris foncé indique une valeur d'appartenance forte.

L'objectif de nos expériences est double. Tout d'abord, il s'agit de tester le comportement de notre approche en fonction de la variation de ses paramètres (granularité des fonctions d'appartenance, formes des fonctions d'appartenance, longueur du support, longueur de la trajectoire dans la fonction d'importance). Ensuite, nous comparerons les résultats obtenus à ceux des autres approches classiques (SIR, AUX-PF, MMPF, AUX-MMPF). En l'absence d'indications contraires, nous utiliserons dans nos tests les paramètres suivants : la famille des fonctions floues correspondant à la vélocité de l'objet est une partition uniforme floue forte de $[-40, 40] \times [-40, 40]$, dont les formes des fonctions sont triangulaires, afin de mettre en valeur le modèle de changements d'états défini par l'approche floue. Le degré d'intersection minimal est fixé à $\epsilon = 0.1$, afin de laisser une probabilité non nulle de passer d'une classe à l'autre. Pour les paramètres du modèle proposé utilisant une fonction de proposition, nous avons utilisé $\gamma = 0.3$ pour l'équation 3.19 afin de favoriser la fonction d'importance, m = 3 pour la longueur de la trajectoire de c et $\beta^1 = \beta^3 = 0.1$ et $\beta^2 = 0.8$ pour utiliser efficacement la dernière observation.

Comme nous allons le constater dans la section 4.2, les valeurs choisies de ces paramètres ne sont pas forcement optimales, mais ont été choisies dans le but de garantir un résultat global satisfaisant dans toutes nos expériences, afin de proposer un jeu de paramètres le moins dépendant possible du test réalisé. Nous proposons ainsi une version simplifiée du modèle, qui est suffisante pour ce qui est montré ici.

Modèles classiques SIR, AUX-PF. Pour les modèles classiques, nous considérons également une dynamique de vélocité constante :

$$\begin{bmatrix} \mathbf{x}_{t} \\ \mathbf{r}_{t} \end{bmatrix} = f_{t}(\mathbf{x}_{t-1}, \mathbf{r}_{t-1}, \mathbf{v}_{t})$$

$$= \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{t-1} \\ \mathbf{r}_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^{2}}{2} & 0 \\ 0 & \frac{\Delta t^{2}}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} \mathbf{v}_{t}^{1} \\ \mathbf{v}_{t}^{2} \end{bmatrix}$$
(4.2)

où $[\mathbf{v}_t^1 \ \mathbf{v}_t^2]^T \sim \mathcal{N}(\mathbf{0}_{2\times 1}, 4^2\mathbf{I}_{2\times 2})$, avec $\mathbf{0}_{2\times 1} = [0 \ 0]^T$ et $\mathbf{I}_{2\times 2}$ la matrice identité 2×2 . Rappelons que, comme signalé dans la section 3.2.1, la prédiction de l'état à l'instant courant fait appel à l'estimation des paramètres à l'instant précédent, à la différence des modèles flous qui prédisent les paramètres avant l'état. Cela signifie que pour les modèles flous, la vraisemblance est évaluée à l'aide de l'état prédit en fonction des paramètres issus du temps t, alors que pour les modèles classiques, ce sont les paramètres à l'instant précédent qui sont utilisés.

Modèles multiples MMPF, AUX-MMPF. Notons c_t la variable aléatoire discrète à valeurs dans $\{1, 2, 3\}$ qui indique l'état dans lequel l'objet se trouve au temps t. Le premier état est une dynamique de vélocité constante, tandis que les deux autres correspondent à des modèles de changements de directions (dans le sens normal et inverse d'une aiguille d'une montre). C'est le modèle qui a été étudié dans [Arulampalam et al., 2004]. Le système s'écrit alors :

$$\begin{bmatrix} \mathbf{x}_t \\ \mathbf{r}_t \end{bmatrix} = f_t(\mathbf{x}_{t-1}, \mathbf{r}_{t-1}, \mathbf{c}_t, \mathbf{v}_t)$$
$$= f_t^{\mathbf{c}_t}(\mathbf{x}_{t-1}, \mathbf{r}_{t-1}, \mathbf{v}_t)$$
(4.3)

avec le modèle à vitesse constante $f_t^1 \triangleq f_t^1(\mathbf{x}_{t-1}, \mathbf{r}_{t-1}, \mathbf{v}_t)$ tel que :

$$f_t^1 = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_{t-1} \\ \mathbf{r}_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} \mathbf{v}_t^1 \\ \mathbf{v}_t^2 \end{bmatrix}$$
(4.4)

avec les modèles de virages $\{f_t^i \triangleq f_t^i(\mathbf{x}_{t-1}, \mathbf{r}_{t-1}, \mathbf{v}_t)\}_{i=2,3}$ tels que :

$$f_{t}^{i} = \begin{bmatrix} 1 & 0 & \frac{\sin(\Omega_{t-1}^{i}\Delta t)}{\Omega_{t-1}^{i}} & -\frac{1-\cos(\Omega_{t-1}^{i}\Delta t)}{\Omega_{t-1}^{i}} \\ 0 & 1 & \frac{1-\cos(\Omega_{t-1}^{i}\Delta t)}{\Omega_{t-1}^{i}} & \frac{\sin(\Omega_{t-1}^{i}\Delta t)}{\Omega_{t-1}^{i}} \\ 0 & 0 & \cos(\Omega_{t-1}^{i}\Delta t) & -\sin(\Omega_{t-1}^{i}\Delta t) \\ 0 & 0 & \sin(\Omega_{t-1}^{i}\Delta t) & \cos(\Omega_{t-1}^{i}\Delta t) \end{bmatrix} \begin{bmatrix} \mathbf{x}_{t-1} \\ \mathbf{r}_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^{2}}{2} & 0 \\ 0 & \frac{\Delta t^{2}}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} \mathbf{v}_{t}^{1} \\ \mathbf{v}_{t}^{2} \end{bmatrix}$$
(4.5)

avec

$$\Omega_{t-1}^2 = \frac{a}{\sqrt{\dot{\mathbf{x}}_{t-1}^2 + \dot{\mathbf{y}}_{t-1}^2}} \tag{4.6}$$

$$\Omega_{t-1}^3 = \frac{-a}{\sqrt{\dot{\mathbf{x}}_{t-1}^2 + \dot{\mathbf{y}}_{t-1}^2}}$$
(4.7)

où $[\mathbf{v}_t^1 \ \mathbf{v}_t^2]^T \sim \mathcal{N}(\mathbf{0}_{2 \times 1}, 4^2 \mathbf{I}_{2 \times 2})$, avec $\mathbf{0}_{2 \times 1} = \begin{bmatrix} 0 \ 0 \end{bmatrix}^T$ et $\mathbf{I}_{2 \times 2}$ la matrice identité 2×2 , et avec a = 50 l'accélération de la manœuvre. La matrice de transition du modèle à saut est donnée sur le tableau 4.2, et privilégie l'état de dynamique linéaire aux dynamiques de changements de direction, qui sont des états de transition.

	1	2	3
1	0.9	0.05	0.05
2	0.4	0.5	0.1
3	0.4	0.1	0.5

TAB. 4.2 – Matrice de transition utilisée pour les approches MMPF et AUX-MMPF. L'indice 1 correspond au modèle de vélocité constante, alors que les indices 2 et 3 indiquent respectivement les modèles de virage dans le sens normal et inverse des aiguilles d'une montre.

4.1.1.2 Modèle de vraisemblance

Dans les séquences synthétiques, le modèle de vraisemblance est linéaire :

$$\mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{r}_t, \mathbf{w}_t) = \mathbf{x}_t + \mathbf{w}_t \tag{4.8}$$

où $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}_{2 \times 1}, \sigma_{\mathbf{w}_t}^2 \mathbf{I}_{2 \times 2})$, avec $\mathbf{0}_{2 \times 1} = (0 \ 0)^T$ et $\mathbf{I}_{2 \times 2}$ la matrice identité 2×2 . La variance du processus de mesure $\sigma_{\mathbf{w}_t}^2$ est variable, et sera explicitée dans les tests.

4.1.2 Séquence de l'hélicoptère

Le but de cette expérience est de valider les approches proposées dans une situation réelle de suivi. Nous ne considérons que le début et la fin de la séquence (l'objet quitte trop fréquemment la scène au milieu de la séquence pour que cette partie soit exploitable). La fin de la séquence n'est considérée que dans les expériences présentées dans la section 4.3.1, et nécessite une réinitialisation manuelle de la position de l'hélicoptère. Elle représente un hélicoptère téléguidé avec une dynamique complexe¹, filmé par une caméra tenue à la main (assez tremblante), ce qui constitue la vraie difficulté pour le suivi (la figure 4.3 présente deux images de la scène).



FIG. 4.3 – (a-b) Deux images de la séquence d'hélicoptère et (c) trajectoire de l'hélicotère dans le repère de l'image.

4.1.2.1 Modèles dynamiques

Nous proposons d'estimer les paramètres de la région ellipsoïdale autour de l'objet, c'est-à-dire les états inconnus de positions et de dynamiques $\mathbf{x}_t = [\mathbf{x}_t \quad \mathbf{y}_t]^T$ et $\mathbf{r}_t = [\dot{\mathbf{x}}_t \quad \dot{\mathbf{y}}_t \quad \theta_t \quad \mathbf{w}_t \quad \mathbf{h}_t]^T$, avec θ_t l'orientation de l'objet, \mathbf{w}_t la largeur et \mathbf{h}_t la hauteur de l'ellipse.

Modèles flous FS-PF, PFS-PF. Le modèle dynamique est celui utilisé dans l'équation 4.1. Les paramètres de vélocité, d'orientation et de dimension sont estimés par notre formalisme flou. Nous considérons également neuf valeurs linguistiques pour le couple $[\dot{\mathbf{x}}_t \ \dot{\mathbf{y}}_t]^T$, de *Nord Ouest* à *Sud Est*, cinq pour l'orientation de l'objet θ_t , de *Nord* à *Sud*, et neuf pour le couple de dimensions de l'ellipse $[\mathbf{w}_t \ \mathbf{h}_t]^T$, de (*PetiteLargeur, PetiteHauteur*) à (*GrandeLargeur, GrandeHauteur*). Ces paramètres sont modélisés d'une manière souple en définissant leurs ensembles flous :

$$\mathbf{c}_{t}^{1} \in \{NO, N, NE, O, Centre, E, SO, S, SE\}$$

$$\mathbf{c}_{t}^{2} \in \{N-S, NE, E, SE\}$$

$$\mathbf{c}_{t}^{3} \in \{(PL, PH), (ML, PH), (GL, PH), \dots, (GL, GH)\}$$

dont les fonctions d'appartenance ont été définies dans la section 2.2.1. Une illustration du paramètre d'orientation est présentée sur la figure 4.4. Notons que le choix d'un paramètre d'orientation à valeurs dans $[-\pi/2, \pi/2]$ et non dans $[0, 2\pi]$ est dû au fait que la vraisemblance que nous allons définir ici est invariante par rotation de π .

¹http://www.youtube.com/watch?v=PQf404RumPE



FIG. 4.4 – Partition uniforme floue du paramètre d'orientation, à valeurs dans $[-\pi/2, \pi/2]$. Les valeurs vont de l'orientation (a) *Nord-Sud* à (d) *Sud-Est*. Un niveau de gris foncé indique une valeur d'appartenance forte.

Modèles classiques SIR, AUX-PF. Le modèle dynamique est celui à vélocité constante présenté dans l'équation 4.2. Comme la vélocité, les paramètres d'orientation et de dimensions suivent une marche aléatoire continue :

$$\begin{bmatrix} \theta_t \\ \mathbf{w}_t \\ \mathbf{h}_t \end{bmatrix} = \begin{bmatrix} \theta_{t-1} \\ \mathbf{w}_{t-1} \\ \mathbf{h}_{t-1} \end{bmatrix} + \begin{bmatrix} \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix} \begin{bmatrix} \mathbf{v}_t^3 \\ \mathbf{v}_t^4 \\ \mathbf{v}_t^5 \end{bmatrix}$$
(4.9)

avec $\{\mathbf{v}_t^i\}_{i=3}^5$ le vecteur de bruit gaussien de moyenne nulle et de matrice de covariance diagonale de paramètres $\{\sigma_{\mathbf{v}_i}^2\}_{i=3}^5$.

Modèles multiples MMPF, AUX-MMPF. Le modèle dynamique est celui présenté dans l'équation 4.3, avec une accélération de la manœuvre fixée de manière empirique à 50. Les paramètres d'orientation et de dimensions suivent des dynamiques de marches aléatoires continues (équation 4.9).

4.1.2.2 Modèle de vraisemblance

Nous modélisons la fonction de vraisemblance comme une combinaison indépendante d'une composante de région (R) et d'une autre de contour (E) :

$$p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{r}_t) = p(\mathbf{y}_t^R | \mathbf{x}_t, \mathbf{r}_t) p(\mathbf{y}_t^E | \mathbf{x}_t, \mathbf{r}_t)$$

où $p(\mathbf{y}_t^E | \mathbf{x}_t, \mathbf{r}_t)$ utilise le gradient de l'image lissée et est défini par :

$$p(\mathbf{y}_t^E | \mathbf{x}_t, \mathbf{r}_t) \propto \exp\left(\frac{\lambda_1}{P_a} \sum_{p=1}^{P_a} \nabla \left[I_t(a^p) \star \mathcal{N}(a^p, \sigma^2)\right]\right)$$
(4.10)

où ∇ et \star sont respectivement les opérateurs de gradient et de convolution, a^p le p-ième point de contour de l'ellipse de longueur P_a et défini par $(\mathbf{x}_t, \mathbf{r}_t)^T$, $I_t(a^p)$ la valeur en niveau de gris du point a^p et de l'image I_t au temps t, $\lambda_1 \in \mathbb{R}^+$ une valeur constante multiplicative fixée qui a pour but de fournir une bonne localisation des fortes valeurs de vraisemblance et $\mathcal{N}(a^p, \sigma^2)$ une distribution normale de moyenne a^p et de variance fixée σ^2 , de valeur assez faible pour ne pas détruire les contours de l'image, tout en lissant légèrement celle-ci afin d'être plus robuste face au bruit.

La vraisemblance de région est donnée par une mesure de distance entre histogrammes. Soit à l'ensemble des pixels couverts par l'ellipse définie par l'état $(\mathbf{x}_t, \mathbf{r}_t)^T$. Nous notons $h_o^{\bullet} = \{h_o^{\bullet}(u)\}_{u=1}^U$ le modèle d'histogramme appris de l'objet (en segmentant l'objet manuellement sur la première image), $h_{BG}^{\bullet} = \{h_{BG}^{\bullet}(u)\}_{u=1}^U$ le modèle d'histogramme appris de l'image de fond (en définissant manuellement une région d'intérêt dans la première image), $h_o(\mathbf{x}_t, \mathbf{r}_t) \triangleq h_o(a) = \{h_o(a; u)\}_{u=1}^U$ un histogramme candidat de l'objet et $h_{BG}(\mathbf{x}_t, \mathbf{r}_t) \triangleq h_{BG}(a) = \{h_{BG}(a; u)\}_{u=1}^U$ un histogramme candidat du fond, avec u l'indice d'une case d'un histogramme de longueur U. L'histogramme candidat du fond contient uniquement les valeurs prises autour de l'objet, et est obtenu par différence entre une dilatation de l'objet de taille 10 et l'objet. La fonction de vraisemblance de base prend en compte uniquement l'histogramme de l'objet candidat et son modèle :

$$p(\mathbf{y}_t^R | \mathbf{x}_t, \mathbf{r}_t) \propto \exp\left(-\lambda_2 d^2 (h_o^{\bullet}, h_o(\mathbf{x}_t, \mathbf{r}_t))\right)$$
(4.11)

avec $\lambda_2 \in \mathbb{R}^+$ une valeur constante multiplicative et $d[h^{\bullet}, h(a)]$ la distance de Bhattacharyya définie par :

$$d[h^{\bullet}, h(\aa)] = \left[1 - \sum_{u=1}^{U} \sqrt{h^{\bullet}(u) h(\aa; u)}\right]^{1/2}$$
(4.12)

L'objectif d'une vraisemblance est d'obtenir une forte valeur uniquement lorsque les valeurs de position, d'orientation ou d'échelle ont correctement été estimées. La version classique de la vraisemblance définie dans l'équation 4.11 suffit généralement pour obtenir une bonne robustesse vis-à-vis de l'orientation et de la position, mais ce n'est pas le cas avec l'échelle, dont les histogrammes sont comparables quelle que soit la valeur de celleci (bien sûr, en considérant des valeurs d'échelle raisonnables), en particulier dans le cas d'objets non texturés (ce qui est notre cas, avec l'hélicoptère) (figures 4.5(C2)(a,b,c,e,f,g)). Ainsi, dans [Pérez et al., 2002], les auteurs proposent de définir une vraisemblance plus robuste par rapport à l'échelle, en utilisant le terme $d^2(h_{BG}^{\bullet}, h_o(\mathbf{x}_t, \mathbf{r}_t))$, dans le but de minimiser la réponse entre le candidat et le modèle d'histogramme de l'image de fond :

$$p(\mathbf{y}_t^R | \mathbf{x}_t, \mathbf{r}_t) \propto \exp\left(-\lambda_2 \left(d^2(h_o^{\bullet}, h_o(\mathbf{x}_t, \mathbf{r}_t)) - d^2(h_{BG}^{\bullet}, h_o(\mathbf{x}_t, \mathbf{r}_t))\right)\right)$$
(4.13)

Le bénéfice par rapport à la version classique (équation 4.11) est évident lorsque l'échelle de l'objet est surestimée. C'est ce que nous pouvons constater sur les figures 4.5(C2)(e,f,g) et 4.5(C3)(e,f,g), qui correspondent à des cas où l'orientation est correcte, mais la valeur de l'échelle est trop forte. Ainsi, cela accroît le terme $d^2(h_{BG}^{\bullet}, h_o(\mathbf{x}_t, \mathbf{r}_t))$, ce qui a pour effet de diminuer la vraisemblance. Malheureusement, lorsque l'échelle est sous-estimée, la vraisemblance proposée par l'équation 4.13 ne permet pas d'obtenir de meilleurs résultats que la méthode classique, à cause de la faible valeur obtenue par le terme $d^2(h_{BG}^{\bullet}, h_o(\mathbf{x}_t, \mathbf{r}_t))$ (figures 4.5(C2)(a,b,c) et 4.5(C3)(a,b,c)). Nous proposons donc de définir une nouvelle vraisemblance, qui, évidemment, fournisse une forte réponse lorsque

le candidat est bien localisé, et une faible dès que l'échelle est mal estimée. Une première idée, qui devrait pénaliser les faibles valeurs d'échelles, est de minimiser le terme $d^2(h_o^{\bullet}, h_{BG}(\mathbf{x}_t, \mathbf{r}_t))$. De cette manière, lorsque l'échelle est trop faible, l'histogramme candidat du fond couvre une partie du vrai objet, ce qui a pour effet d'augmenter le terme $d^2(h_o^{\bullet}, h_{BG}(\mathbf{x}_t, \mathbf{r}_t))$. Cette vraisemblance peut se définir de cette manière :

$$p(\mathbf{y}_t^R | \mathbf{x}_t, \mathbf{r}_t) \propto \exp\left(-\lambda_2 \left(d^2(h_o^{\bullet}, h_o(\mathbf{x}_t, \mathbf{r}_t)) - d^2(h_o^{\bullet}, h_{BG}(\mathbf{x}_t, \mathbf{r}_t))\right)\right)$$
(4.14)

Les résultats sont illustrés sur la figure 4.5(C4). Comme nous pouvions l'espérer, son comportement lorsque l'échelle est trop faible est correct. En revanche, lorsqu'elle est surevaluée, le terme $d^2(h_o^{\bullet}, h_{BG}(\mathbf{x}_t, \mathbf{r}_t))$ est faible, car l'histogramme candidat du fond ne couvre plus l'objet. L'idée consiste donc à cumuler les avantages des vraisemblances proposées dans les équations 4.13 et 4.14, en considérant la valeur maximale entre les deux termes et $d^2(h_{BG}^{\bullet}, h_o(\mathbf{x}_t, \mathbf{r}_t))$ et $d^2(h_o^{\bullet}, h_{BG}(\mathbf{x}_t, \mathbf{r}_t))$, dans le but de pénaliser les situations où l'échelle est trop faible, ou trop forte :

$$p(\mathbf{y}_{t}^{R}|\mathbf{x}_{t},\mathbf{r}_{t}) \propto \exp\left(-\lambda_{2}\left(d^{2}(h_{o}^{\bullet},h_{o}(\mathbf{x}_{t},\mathbf{r}_{t}))-\max\left[d^{2}(h_{BG}^{\bullet},h_{o}(\mathbf{x}_{t},\mathbf{r}_{t})),d^{2}(h_{o}^{\bullet},h_{BG}(\mathbf{x}_{t},\mathbf{r}_{t}))\right]\right)\right)$$
(4.15)

Comme nous pouvons le constater sur la figure 4.5(C5), cette vraisemblance fournit une réponse robuste aux changements d'échelle, ce qui était souhaité.

En suivant les choix faits par [Pérez et al., 2002], nous utilisons une représentation de l'espace de couleurs en Teinte Saturation Valeur (TSV, ou HSV en anglais, pour *Hue Saturation Value*, ou HSB, pour *Hue Saturation Brightness*), et pour chaque pixel, nous considérons soit la valeur du couple teinte et saturation si la saturation et la brillance ont des valeurs plus grandes que des seuils fixés à 0.1 et 0.2, respectivement, soit la brillance du pixel. Les paramètres ont été fixés aux valeurs $U_t = U_s = U_v = 10$ et $\lambda_1 = \lambda_2 = 10$. Utiliser une quantification de $U_* = 10$ valeurs par dimension permet de représenter de manière compacte la distribution des valeurs de l'image dans un histogramme, et ainsi être plus robuste au bruit. Les constantes multiplicatives λ_* permettent de localiser assez précisément les fortes concentrations de la vraisemblance, ce qui permet une plus forte discrimination des particules lors de la mise à jour des poids dans l'algorithme du filtre particulaire. Enfin, notons que notre implantation du calcul d'histogramme et de la comparaison d'histogrammes est naïve, et pourrait être améliorée en utilisant une approche plus optimisée [Dubuisson, 2010; Porikli, 2005].

4.1.3 Séquence de bouche

Dans la littérature du filtre particulaire, il existe peu de filtres intégrant un modèle de forme évolué. Bien souvent, les applications ne modélisent que des formes basiques, telles que des rectangles ou des ellipses, dans le but de limiter la dimension du vecteur à estimer, et ainsi éviter le problème lié à la dimension de l'échantillonnage préférentiel (sections 1.1.2.3 et 1.3.1.4). Toutefois, nous pouvons énumérer quelques travaux intégrant des informations de formes de plus haut niveau. Le premier en date est celui proposé par [Isard et Blake, 1998a], dans lequel la dynamique de la forme est apprise selon un processus auto-regressif du second ordre. Nous reviendrons sur ce modèle plus tard. Dans [Chang et al., 2008], les auteurs proposent de guider le suivi de formes par filtre particulaire en utilisant des modèles d'apparence issus d'une base d'apprentissage, puis



FIG. 4.5 – Log-vraisemblances obtenues en chaque point de l'image, selon les formes représentées en colonne C1. Une couleur rouge indique une forte valeur de correspondance, tandis qu'une couleur virant au bleu en indique une faible. (C2) vraisemblance classique 4.11, (C3) vraisemblance 4.13 proposée par [Pérez et al., 2002], (C4) vraisemblance 4.14 et (C5) vraisemblance 4.15. Les lignes correspondent à des échelles différentes, (a) échelle 0.1, (b) échelle 0.4, (c) échelle 0.7, (d) échelle correcte 1, (e) échelle 1.3, (f) échelle 1.6 et (b) échelle 1.9. L'objectif est d'obtenir une forte vraisemblance (c'est-àdire, une couleur rouge vive) lorsque le pixel considéré est au centre de l'objet et lorsque l'échelle est correcte (ligne (d)). Lorsqu'une de ces conditions n'est pas vérifiée, nous souhaitons obtenir une vraisemblance faible. La meilleure vraisemblance semble celle proposée dans la colonne C5, puisque les valeurs de vraisemblance sont faibles lorsque la position ou l'échelle sont mal estimées, et restent fortes lorsque celles-ci sont correctes (ligne (d), centre de l'objet).

en déformant les formes candidates par transformations affines. Le principal défaut de ces approches est l'utilisation d'un modèle dynamique linéaire, qui ne peut pas gérer les déformations locales d'une forme. Ainsi, dans [Rathi et al., 2007], les auteurs introduisent des formes apprises représentées par des ensembles de niveaux, et pour la procédure de suivi, proposent des nouvelles formes en procédant à une descente de gradient sur l'énergie associée à la forme décrite par la particule à l'instant précédent. Cependant, cette approche possède un coût de calcul élevé, puisque chaque particule est liée à un ensemble de niveaux. De plus, la taille de la base de données doit être conséquente pour intégrer la plupart des variations de la forme.

Comme nous l'avons vu dans la section 2.3, notre approche floue possède l'avantage de représenter une forme sans déterminer de modèle physique. Les formes apprises sont entièrement conditionnées par la base d'apprentissage et les contraintes de formes que nous souhaitons préserver. Cela permet de manipuler des formes implicites, qui sont représentées par des ensembles flous. La dynamique entre ces ensembles, que nous appelons dans ce cas postures, est calculée en utilisant un degré d'intersection. Bien que, comme nous allons le découvrir, il est difficile de faire une comparaison directe entre plusieurs filtres particulaires intégrant une information de forme, nous avons choisi de comparer notre approche à celle proposée par [Isard et Blake, 1998a], qui possède l'avantage de mêler efficacité en termes de performances et de coût de calculs. Dorénavant, nous nommerons cette approche Isard *et al.*.

Dans la section 4.4, nous présenterons donc une application de suivi de formes. Des illustrations tirées de la séquence sont données sur la figure 4.6. 7 images par classes (donc 21 en tout) ont été considérées comme base d'apprentissage, et 180 comme base de tests (les 180 premières images de la séquence). Seul le modèle flou sans fonction de proposition (FS-PF) est considéré dans cette expérience, afin de mettre en évidence l'apport de notre approche en termes de modélisation et non de simulation.



FIG. 4.6 – Images de la séquence de suivi de bouche. Postures des lèvres (a) *fermée*, (b) *ouverte* et (c) *sourire*.

4.1.3.1 Modèle dynamique

Modèle flou FS-PF. Le vecteur d'état $\mathbf{x}_t = (\mathbf{x}_t, \mathbf{y}_t)$ contient les coordonnées 2D du centre de la bouche au temps t. Le vecteur de paramètres $\mathbf{r}_t = (\dot{\mathbf{x}}_t, \dot{\mathbf{y}}_t, \theta_t, \mathbf{a}_t)$ intègre le vecteur de déplacement $(\dot{\mathbf{x}}_t, \dot{\mathbf{y}}_t)$, l'orientation de l'objet θ_t et un vecteur de points de contrôle $\mathbf{a}_t = (\mathbf{a}_t^1, \ldots, \mathbf{a}_t^P)$. Nous considérons le modèle dynamique présenté dans l'équation 4.1. Le vecteur de bruit de position suit $[\mathbf{v}_t^1 \ \mathbf{v}_t^2]^T \sim \mathcal{N}(\mathbf{0}_{2\times 1}, 2^2\mathbf{I}_{2\times 2})$, avec $\mathbf{0}_{2\times 1} = (0\ 0)^T$, $\mathbf{I}_{2\times 2}$ la matrice d'identité 2×2 .

Les paramètres de vélocité, d'orientation et de forme de bouche sont modélisés de

manière souple en définissant leurs valeurs possibles par des ensembles flous :

$$\begin{array}{rcl} \mathbf{c}_{t}^{1} & \in & \{NO, N, NE, O, Centre, E, SO, S, SE\} \\ \mathbf{c}_{t}^{2} & \in & \{N, NE, E, SE, S\} \\ \mathbf{c}_{t}^{3} & \in & \{Ferm \acute{e}, Ouverte, Sourire\} \end{array}$$

dont les fonctions d'appartenance ont été définies dans la section 2.2.1. Ainsi, le vecteur de déplacement peut appartenir à neuf ensembles flous, du *Nord-Ouest* au *Sud-Est*, l'orientation de l'objet, du *Nord* au *Sud*, et les trois ensembles de points de contrôle aux postures floues *Fermée*, *Ouverte* et *Sourire*.

Nous modélisons la famille de fonctions floues correspondant à la vélocité de l'objet comme une partition uniforme floue forte de $[-20, 20] \times [-20, 20]$, et celle de l'orientation comme une partition uniforme floue forte de $[0, 2\pi]$. La forme des fonctions floues pour les paramètres de vélocité et d'orientation est triangulaire.

La famille des fonctions de posture de la bouche est obtenue automatiquement par apprentissage. La probabilité *a priori* des paramètres est approchée selon l'algorithme de Metropolis-Hasting (section 3.6). Le calcul du degré d'intersection défini dans la section 2.3.2.2 s'écrit alors :

$$\begin{split} \Xi(\mu_k^3, \mu_h^3) &= \frac{\int_{\mathcal{R}} \mu_{\mathcal{Z}}(\mathbf{r}) \Upsilon \left[\mu_k^3(\mathbf{r}), \mu_h^3(\mathbf{r}) \right] d\mathbf{r}}{\min \left[\int_{\mathcal{R}} \mu_{\mathcal{Z}}^3(\mathbf{r}) \mu_k^3(\mathbf{r}) d\mathbf{r}, \int_{\mathcal{R}} \mu_{\mathcal{Z}}^3(\mathbf{r}) \mu_h^3(\mathbf{r}) d\mathbf{r} \right]} \\ &= \frac{\int_{\mathcal{R}} \mathbb{P}(\mathbf{r} | \mathcal{Z}^3) \Upsilon \left[\mu_k^3(\mathbf{r}), \mu_h^3(\mathbf{r}) \right] d\mathbf{r}}{\min \left[\int_{\mathcal{R}} \mathbb{P}(\mathbf{r} | \mathcal{Z}^3) \mu_k^3(\mathbf{r}) d\mathbf{r}, \int_{\mathcal{R}} \mathbb{P}(\mathbf{r} | \mathcal{Z}^3) \mu_h^3(\mathbf{r}) d\mathbf{r} \right]} \\ &\approx \frac{\sum_{l=1}^L \Upsilon \left[\mu_k^3(\mathbf{r}^{(l)}), \mu_h^3(\mathbf{r}^{(l)}) \right]}{\min \left[\sum_{l=1}^L \mathbb{1}^k(\mathbf{r}^{(l)}), \sum_{l=1}^L \mathbb{1}^h(\mathbf{r}^{(l)}) \right]} \end{split}$$

avec $\Upsilon(a, b) = 1$ si $a \times b \neq 0$ et 0 sinon, et $\mathbb{1}^k(x) = \Upsilon(\mu_k(x), 1)$ la fonction indicatrice de la classe k.

Modèle proposé par Isard *et al.*. Le modèle dynamique de formes proposé dans [Isard et Blake, 1998a] est un processus auto-régressif du second ordre :

$$\mathbf{x}_{t} = A_{2}\mathbf{x}_{t-2} + A_{1}\mathbf{x}_{t-1} + D_{0} + B_{0}\mathbf{w}_{t}$$
(4.16)

avec \mathbf{w}_t un vecteur de variables aléatoires indépendantes de lois normales, \mathbf{x}_t un vecteur aléatoire représentant la forme de l'objet à valeurs dans \mathbb{R}^{N_x} , D_0 un vecteur de décalage de dimension N_x , A_1 et A_2 des matrices de dimensions $N_x \times N_x$ représentant la composante dynamique déterministe du modèle, et B_0 de dimensions $N_x \times N_x$ le coefficient de covariance.

Un tel modèle avait déjà été proposé dans [Blake et al., 1995] et [Blake et Isard, 1998] (chapitre 9), mais il fallait alors définir manuellement les paramètres A_1 , A_2 , D_0 et B_0 , ce qui peut rapidement se montrer fastidieux et non intuitif dans des cas complexes. La solution adoptée dans [Isard et Blake, 1998a], [Isard, 1998] (chapitre 2) ou encore [Doucet et al., 2001a] (chapitre 16) consiste à apprendre, grâce à une base de données, les paramètres du modèle dynamique 4.16 par une méthode de maximum de vraisemblance. L'algorithme d'apprentissage est décrit dans [Isard, 1998] (chapitre 2), et est donné pour rappel dans l'algorithme 4.1.

Pour bien fonctionner, cette méthode nécessite un grand nombre d'échantillons dans la base de données, couvrant, si possible, toutes les variations de la forme qu'il est possible d'observer. Dans notre application de suivi de bouche, nous avons pris 1000 formes. Cette quantité peut être un frein à la mise en œuvre du filtre. C'est un inconvénient dont notre modèle flou ne souffre pas, puisque seules 21 images ont été utilisées pour générer les fonctions d'appartenance de la bouche.

Algorithme 4.1: Algorithme d'apprentissage des paramètres d'un processus autorégressif du second ordre, tiré de la thèse [Isard, 1998] (chapitre 2).

- **Entrées** : Un ensemble de *M* formes centrées issues de la base d'apprentissage $\{\mathbf{z}_1, \dots, \mathbf{z}_M\}$
- **Sorties** : Les paramètres du processus auto-régressif du second ordre A_1 , A_2 , D_0 et B_0 .

début

1. Calculer les sommes R_i , i = 0, 1, 2 et les coefficients d'auto-corrélation R_{ij} et R'_{ij} , i, j = 0, 1, 2:

$$R_{i} = \sum_{k=3}^{M} \mathbf{z}_{k-i}$$

$$R_{ij} = \sum_{k=3}^{M} \mathbf{z}_{k-i} \mathbf{z}_{k-j}^{T}$$

$$R'_{ij} = R_{ij} - \frac{1}{M-2} R_{i} R_{j}^{T}$$

2. Estimer les paramètres A_1 , A_2 et D_0 :

$$A_{2} = \left(R_{02}' - R_{01}' R_{11}'^{-1} R_{12}'\right) \left(R_{22}' - R_{21}' R_{11}'^{-1} R_{12}'\right)^{-1}$$

$$A_{1} = \left(R_{01}' - A_{2} R_{21}'\right) R_{11}'^{-1}$$

$$D_{0} = \frac{1}{M - 2} \left(R_{0} - A_{2} R_{2} - A_{1} R_{1}\right)$$

3. Estimer le paramètre B_0 comme la matrice carré $B_0 = \sqrt{C}$:

$$C = \frac{1}{M-2} \left(R_{00} - A_2 R_{20} - A_1 R_{10} - D_0 R_0^T \right)$$

retourner A_1 , A_2 , D_0 et B_0 .

4.1.3.2 Modèle de vraisemblance

En suivant le choix effectué dans la section 4.1.2, nous utilisons une vraisemblance combinant des informations de contour et de région :

$$p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{r}_t, \mathbf{c}_t^3 = k) = p(\mathbf{y}_t^R | \mathbf{x}_t, \mathbf{r}_t, \mathbf{c}_t^3 = k) p(\mathbf{y}_t^C | \mathbf{x}_t, \mathbf{r}_t)$$

où $p(\mathbf{y}_t^C | \mathbf{x}_t, \mathbf{r}_t)$ correspond à la vraisemblance de contour définie dans l'équation 4.10, calculée le long de la B-spline interpolée des points de contrôle (a_t^1, \ldots, a_t^P) . La vraisem-

blance de région, $p(\mathbf{y}_t^R | \mathbf{x}_t, \mathbf{r}_t, \mathbf{c}_t^3 = k)$, est conditionnée par rapport à la posture, ici k. Cela permet de modéliser une vraisemblance par posture, dans le but d'enrichir le modèle d'observation, et donc d'améliorer la robustesse du suivi. Malheureusement, par nature, ce modèle ne peut être calqué sur l'approche d'Isard *et al.*, qui n'utilise pas la variable c. Ainsi nous sommes obligés de définir deux modèles de vraisemblance différents, c'est pourquoi la comparaison de notre approche et celle d'Isard *et al.*ne peut se faire directement. Les résultats observés seront donc en partie conditionnés par cette différence.

Modèle flou FS-PF. La vraisemblance utilisant les régions prend en compte les indices des classes \mathbf{c}_t . Elle est liée, d'une part à une distance entre l'histogramme d'un modèle de bouche $h_k^{\bullet} = \{h_k^{\bullet}(u)\}_{u=1}^U$ de posture k, et un histogramme candidat $h(a_t) = \{h(a_t; u)\}_{u=1}^U$ [Pérez et al., 2002], avec u l'indice d'une case d'un histogramme de longueur U, et d'autre part à une distance entre l'histogramme candidat et l'histogramme $h_0^{\bullet} = \{h_0^{\bullet}(u)\}_{u=1}^U$ de l'arrière plan de l'image :

$$p(\mathbf{y}_{t}^{R}|a_{t}, \mathbf{c}_{t}^{3} = k) \propto \exp\left(-\lambda_{2}(d^{2}[h_{k}^{\bullet}, h(a_{t})] - d^{2}[h_{0}^{\bullet}, h(a_{t})])\right)$$
(4.17)

où les histogrammes de référence $\{h_k^{\bullet}\}_{k=0}^3$ sont créés pendant la phase d'apprentissage. Pour k = 1, 2, 3, les histogrammes sont calculés comme la moyenne des histogrammes obtenus en utilisant plusieurs lèvres ségmentées manuellement et choisies aléatoirement à partir de la base d'apprentissage. Pour k = 0, l'histogramme correspond à la moyenne des histogrammes obtenus en utilisant plusieurs images de fond issues de la base d'apprentissage.

Modèle proposé par Isard *et al.*. Le modèle de vraisemblance que nous allons définir ici est le modèle classique, c'est-à-dire sans utilisation de la variable discrète c^3 , défini dans l'équation 4.17. Il correspond alors à une loi de mélange, comportant autant de composantes que c^3 possède de valeurs différentes. Chaque composante indique alors une posture, dont les probablités sont notées $\{\pi_k\}_{k=1}^3$:

$$p(\mathbf{y}_{t}^{R}|a_{t}) = \sum_{k=1}^{3} \pi_{k} \, p(\mathbf{y}_{t}^{R}, a_{t}, \mathbf{c} = k)$$
(4.18)

avec $p(\mathbf{y}_t^R, a_t, \mathbf{c} = k)$ la vraisemblance décrite dans l'équation 4.17. Les paramètres de la loi de mélange sont uniformes, donc $\pi_1 = \pi_2 = \pi_3 = 1/3$.

Pour les vraisemblances 4.17 et 4.18, nous avons choisi un espace colorimétrique décomposé en teinte T, saturation S et valeur V. L'histogramme est de longueur $U = U_T \times U_S \times U_V$, avec $U_T = U_S = U_V = 10$ le nombre de cases par composantes. Les constantes multiplicatives λ_1 et λ_2 ont été fixées à 20.

4.2 Influence des paramètres

Nous étudions dans cette section l'influence des paramètres de notre modèle sur les résultats obtenus, en utilisant les séquences synthétiques définies dans la section 4.1.1. La section 4.2.1 décrit les expériences sur le modèle flou proposé, sans l'utilisation de la fonction de proposition. Les classes sont donc générées selon l'équation 3.16, alors que la section 4.2.2 traite de l'influence des paramètres dans la fonction de proposition. Nous

nous intéressons plus au comportement global des modèles proposés vis-à-vis des paramètres qu'à l'étude minutieuse des performances en termes d'erreurs obtenues, puisqu'il faut garder en perspective le fait que les séquences utilisées ici sont synthétiques, donc d'une difficulté faible, ce qui favorise particulièrement les implantations naïves. Un récapitulatif des tests réalisés dans cette section est proposé dans le tableau 4.3.

Méthodes	Séquences	Tests	Sections
		Granularité	4.2.1.1
FS-PF	Synthétiques 1,2,3	Formes des fonctions	4.2.1.2
		Longueur du support	4.2.1.3
PFS-PF	Synthétiques 1,2,3	Longueur de la trajectoire	4.2.2.1

TAB. 4.3 – Récapitulatif des méthodes, séquences et tests étudiés dans la section 4.2.

4.2.1 Modèle flou

4.2.1.1 Granularité de la famille de fonctions d'appartenance

Nous étudions dans cette section l'influence de la granularité choisie de la famille de fonctions d'appartenance. La longueur du support de la partition uniforme floue forte de la vélocité est fixée, et la forme des fonctions floues est triangulaire. La figure 4.7 présente les résultats en fonction du nombre de particules, en utilisant 1, 3, 5, 7 et 9 fonctions floues par dimension, soit au total 1, 9, 25, 49 et 81 fonctions d'appartenance. En complément, la figure 4.8 présente l'erreur obtenue en fonction du nombre de fonctions d'appartenance par dimension. Nous pouvons tout d'abord constater la robustesse du modèle vis-à-vis de ce paramètre, puisque les résultats restent stables, et ce, pour n'importe quel nombre de particules considéré (figure 4.8). Toutefois, dans la séquence linéaire, utiliser 3 ou 5 fonctions d'appartenance par dimension semble être plus profitable (figure 4.7(a)). Cela s'explique certainement par l'introduction du modèle de transition souple défini, qui favorise les linéarités. Dans les séquences erratiques et classique, le résultat en ne partitionnant pas l'espace semble légèrement meilleur, lorsque le nombre de particules est faible. Cela se comprend par la nature des mouvements, qui ne permettent pas d'utiliser efficacement la notion de linéarité du modèle (rappel : une classe a toujours une probabilité maximale de rester dans le même état). Comme nous le verrons dans la section 4.3, la fonction de proposition permet justement de facilemement dépasser cette limite, en utilisant la dernière observation pour générer les particules, et ainsi gérer les dynamiques changeantes. En outre, rappelons que les expériences réalisées ici ne modélisent que le paramètre de vélocité, et agissent donc dans un espace de simulation restreint; l'intérêt d'un modèle évolué sur un modèle uniforme devient évident dans des applications réelles, tant une simple simulation uniforme souffre très rapidement d'une augmentation de la dimension du vecteur de la dynamique (sections 4.3.1.3 et 4.3). En conclusion, au vu des résultats dans les trois scénarios, les choix de 3 ou 5 fonctions d'appartenance par dimension semblent être pertinent.

4.2.1.2 Formes des fonctions d'appartenance et paramètre ϵ

Nous faisons maintenant varier la forme des fonctions d'appartenance (triangulaire, trapézoïdale et rectangulaire) ainsi que le paramètre de degré minimal d'intersection


FIG. 4.7 – Granularité : erreurs obtenues en fonction du nombre de particules. Les courbes représentées indiquent le nombre de fonctions d'appartenance considéré par dimension. (a) séquence linéaire (scénario synthéthique 1), (b) séquence erratique (scénario synthéthique 2) et (c) séquence classique (scénario synthéthique 3).



FIG. 4.8 – Granularité : erreurs obtenues en fonction du nombre de fonctions d'appartenance par dimension. Les courbes représentées indiquent le nombre de particules considéré. (a) séquence linéaire (scénario synthéthique 1), (b) séquence erratique (scénario synthéthique 2) et (c) séquence classique (scénario synthéthique 3).

entre deux classes ϵ (équation 2.4). La longueur du noyau des fonctions trapézoïdales est fixée à la moitié de celle du noyau des fonctions rectangulaires (figure 4.9).



FIG. 4.9 – Fonctions d'appartenance de formes (a) rectangulaire, (b) trapézoïdale, et (c) triangulaire. Toutes les fonctions ont une aire identique.

La figure 4.10 présente les erreurs obtenues en fonction du nombre de particules, et la figure 4.11 en fonction du degré minimal d'intersection. D'autres tests ont été effectués dans l'annexe B.1. Les formes triangulaires et trapézoïdales donnent des erreurs stables en fonction du paramètre ϵ . Une forme triangulaire et une valeur de paramètre $\epsilon = 0.1$ semblent fournir des bons résultats dans tous nos tests (voir l'annexe B.1). Dans les tests classiques et erratiques, la valeur $\epsilon = 0.1$ permet d'obtenir un résultat quasiment équivalent de celui obtenu avec $\epsilon = 1$, qui correspond à une recherche uniforme dans l'espace des paramètres. La raison est la même que celle explicitée dans la section 4.2.1.1, c'est-àdire que de par la nature erratique de ces expériences, un modèle linéaire ne peut faire mieux qu'un modèle uniforme.



FIG. 4.10 – Tests sur les formes des fonctions d'appartenance : erreurs obtenues en fonction du nombre de particules, de la forme des fonctions floues (triangulaire, trapézoïdale et rectangulaire) et avec un degré minimal d'intersection entre deux classes $\epsilon = 0.1$. (a) Séquence linéaire (scénario synthéthique 1), (b) séquence erratique (scénario synthéthique 2) et (c) séquence classique (scénario synthéthique 3).



FIG. 4.11 – Tests sur les formes des fonctions d'appartenance : erreurs obtenues en fonction du degré minimal d'intersection ϵ entre deux classes, de la forme des fonctions floues (triangulaire, trapézoïdale et rectangulaire) et avec un nombre de particules N = 20. (a) Séquence linéaire (scénario synthéthique 1), (b) séquence erratique (scénario synthéthique 2) et (c) séquence classique (scénario synthéthique 3).

4.2.1.3 Longueur du support

Nous faisons maintenant varier la longueur du support du paramètre de vélocité. Ce paramètre définit une partition floue uniforme forte sur $[-h_x/2, h_x/2] \times [-h_y/2, h_y/2]$. Nous considérons ici les longueurs telles que $h_x = h_y$, et dont nous faisons varier la valeur. La figure 4.12 présente les résultats obtenus en fonction du nombre de particules et la figure 4.13 présente ceux obtenus en fonction de la longueur du support. Dans les trois expériences, nous pouvons constater que la plage de valeurs [50, 100] permet d'obtenir des résultats concluants, et donne rapidement une même erreur minimale lorsque le nombre de particules augmente. Le choix du paramètre nécessite une information *a priori* assez faible, et permet ainsi d'obtenir facilement des résultats satisfaisants. En particulier, la valeur d'horizon h = 80 utilisée dans nos expériences semble adéquate.

4.2.2 Fonction d'importance

Dans cette section, nous étudions l'influence des paramètres utilisés dans la fonction d'importance définie dans la section 3.4.

4.2.2.1 Longueur de la trajectoire : paramètre *m*

Tout faisons tout d'abord varier la valeur de la longueur de la trajectoire m. Pour ne prendre en compte que la trajectoire, les paramètres $\{\alpha_k\}_{k=1}^{K}$ (quantités multiplicatives des modes de la loi *a priori* de Dirichlet) sont fixés à 0. Mettre en évidence l'intérêt d'utiliser la trajectoire dans la simulation des particules n'est pas direct, puisqu'il s'agit de situations complexes, dans lesquelles l'information à t - 1 ne suffit pas. Ainsi, nous pro-



FIG. 4.12 – Longueur du support : erreurs obtenues en fonction du nombre de particules et de la longueur du support considérée dans chaque dimension. (a) Séquence linéaire (scénario synthéthique 1), (b) séquence erratique (scénario synthéthique 2) et (c) séquence classique (scénario synthéthique 3).



FIG. 4.13 – Longueur du support : erreurs obtenues en fonction de la longueur du support considérée dans chaque dimension et du nombre de particules. (a) Séquence linéaire (scénario synthéthique 1), (b) séquence erratique (scénario synthéthique 2) et (c) séquence classique (scénario synthéthique 3).

posons dans cette section deux scénarios pour lesquels l'utilisation de la trajectoire est claire.

Le premier est illustré sur la figure 4.14(a), et représente une trajectoire parfaitement linéaire. Le modèle de mesures est le suivant : le capteur a une probabilité de 0.7 de fournir une mesure correcte (selon le modèle de mesure défini, ici $\mathbf{y}_t = \mathbf{x}_t + \mathbf{w}_t$, avec $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}_{2\times 1}, 4^2 \mathbf{I}_{2\times 2}))$, et une probabilité de 0.3 de fournir une mesure erronée (générée selon une loi uniforme et éloignée du vrai état). Cela est représenté par les mesures colorées en vert dispersées autour de la vraie trajectoire sur la figure 4.14(a). Les résultats obtenus en fonction du nombre de particules et de la longueur de la trajectoire sont illustrés sur les figures 4.14(b) et 4.14(c). Une longueur de trajectoire égale à 0 indique que les particules ont été générées selon la transition *a priori* (équation 3.16), ou, dit différemment, $\gamma = 1$, avec $\epsilon = 0.1$. Pour toutes les autres valeurs de la longueur de la trajectoire, nous fixons $\gamma = 0$, afin de prendre en compte uniquement la trajectoire. Les résultats montrent le bénéfice d'utiliser une longueur de la trajectoire grande. Lorsque $m \ge 13$, les résultats obtenus deviennent même meilleurs qu'en utilisant le modèle de transition classique, alors que celui-ci privilégie les linéarités. Cela signifie qu'au-delà d'une longueur de trajectoire égale à 13, la probabilité inférée par la fonction d'importance de considérer le même état devient plus importante que celle définie par la matrice de transition a priori. Nous pouvons également avoir l'intuition de ce phénomène de manière plus formelle : la vélocité étant définie par neuf ensembles flous et le déplacement dans cette séquence ne s'effectuant que vers le Nord-Est (cela n'est pas forcément vrai, mais simplifions), le vecteur de paramètres de la loi de Dirichlet est donc q = (1, 1, 14, 1, 1, 1, 1, 1, 1). L'espérance de la loi de Dirichlet pour la valeur Nord-Est (par marginalisation) est égale à q^{N-E}/q^0 , avec q^{N-E} le nombre de fois où la valeur N-E a été considérée, $q^0 \triangleq \sum_{k=1}^{K} q^i$, soit $14/22 \simeq 0.64$. Or, cette probabilité est plus forte que celle calculée automatiquement via l'estimation du degré d'intersection entre les ensembles flous (section 2.2.2), qui vaut dans le cas présent $1/(1 + 0.25 * 2 + 0.125) \simeq 0.61$. Ainsi, plus la longueur de la trajectoire est grande, plus la probabilité de considérer la valeur Nord-Est est forte, ce qui est représenté sur la figure 4.14(d).

Dans le second test (figure 4.15(a)), nous considérons une séquence générée avec un modèle de vélocité uniforme, ce qui explique l'aspect totalement erratique de la trajectoire. Les observations ont été simulées selon le modèle classique de mesure, $\mathbf{y}_t = \mathbf{x}_t + \mathbf{w}_t$, avec $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}_{2\times 1}, 4^2\mathbf{I}_{2\times 2})$. L'intérêt d'utiliser la trajectoire est ici évident, puisque la fonction d'importance a alors tendance à inférer un modèle à sauts uniforme, c'est pourquoi l'erreur obtenue ne cesse de décroître.

4.2.2.2 Discussion à propos des autres paramètres

La fonction d'importance utilise d'autres paramètres, qui sont $\{\alpha_k\}_{k=1}^K, \{\beta^i\}_{i=1,2,3}$ et γ . Toutefois, l'étude de l'influence de ces paramètres ne semble pas réellement pertinente, puisque leurs valeurs sont plutôt fixées par le bon sens. Les paramètres $\{\alpha_k\}_{k=1}^K$ servent à pondérer l'importance des modes, et généralement, ceux-ci étant tous d'égale importance, $\forall k, \alpha_k = \alpha = cste$. Ainsi, la constante α sert à pondérer l'utilisation de la trajectoire par rapport à celle des nouvelles probabilités des modes, et peut donc par exemple être fixée telle que $\alpha = 2m$, afin de privilégier les nouvelles probabilités.

Les valeurs $\{\beta^i\}_{i=1,2,3}$ sont les paramètres de la loi de mélange des nouvelles probabilités des modes. Là encore, fixer une forte valeur pour le paramètre associé à la probabilité dépendant de la nouvelle observation semble tout à fait indiqué. Toutefois, la valeur de



FIG. 4.14 – (a) Séquence parfaitement linéaire, mais dont les observations peuvent être erronées, (b) erreurs obtenues en fonction du nombre de particules, (c) erreurs obtenues en fonction de la longueur de la trajectoire et (d) espérance de la loi de Dirichlet de paramètres (1, 1, x, 1, 1, 1, 1, 1, 1) obtenue par la variable *Nord-Est*.



FIG. 4.15 - (a) Séquence générée avec un modèle de vélocité uniforme, (b) erreurs obtenues en fonction du nombre de particules et (c) erreurs obtenues en fonction de la longueur de la trajectoire.

ces paramètres pourrait être mise en relation avec la valeur de bruit du modèle de mesure, ou encore, dans le cas où l'information *a priori* serait essentielle (ce qui pourrait être le cas avec une loi *a priori* atypique), une forte valeur de β^1 semblerait indiquée. Cependant, dans nos tests, nous considérons que l'information de l'observation est primordiale, et nous fixons ainsi le jeu de paramètres tel que $\beta^1 = \beta^3 = 0.1$ et $\beta^2 = 0.8$.

Enfin, le paramètre γ pondère l'utilisation de la fonction d'importance par rapport à celle de la transition *a priori*. La valeur de ce paramètre ne semble pas dépendre de l'application, et peut logiquement être choisie afin de favoriser l'utilisation de la fonction d'importance (dans nos tests, nous fixons sa valeur à 0.7).

4.3 Les relations spatiales floues en pratique

Nous comparons dans cette section les approches proposées (FS-PF et PFS-PF) aux méthodes classiques (SIR, AUX-PF) et aux méthodes à modèles dynamiques multiples (MMPF, AUX-MMPF), dans une problématique de suivi d'objets classique, faisant appel aux relations spatiales floues. L'objectif est de montrer le comportement des modèles flous dans divers scénarios, en les comparant à d'autres modèles, ainsi que l'intérêt de la fonction de proposition de la méthode PFS-FS par rapport au modèle flou simple FS-PF. Dans la section 4.3.1, nous étudierons l'adaptabilité des modèles dans des conditions de suivi extrêmes avant de mettre l'accent sur la robustesse et le temps de calcul dans la section 4.3.2. Un récapitulatif des expériences réalisées dans cette section est proposé dans le tableau 4.4.

Séquences	Expériences	Sections		
Synthétique 1	Adaptabilité	4.3.1.1		
Synthétique 2	Synthétique 2 Adaptabilité			
Synthétique 3	Temps de calcul / Robustesse	4.3.2.1		
Hélicoptère	Adaptabilité	4.3.1.3		
	Temps de calcul / Robustesse	4.3.2.2		

TAB. 4.4 – Récapitulatif des séquences et expériences étudiées dans la section 4.3.

4.3.1 Adaptabilité du modèle

L'approche proposée modélise la dynamique des paramètres par une densité de probabilité uniforme. Cependant, elle intègre une décomposition floue de l'espace des paramètres, en utilisant une information spatiale, pour favoriser les régions de l'espace des paramètres qui semblent contenir les vraies valeurs des paramètres dynamiques. Ainsi, même si la densité de probabilité considérée est uniforme, le modèle flou injecte l'idée de linéarité du modèle à vélocité constante, en donnant une probabilité maximale à une classe d'être considérée de nouveau à l'instant suivant.

Cette modélisation peut être vue comme une manière implicite d'intégrer une information de vélocité, ce qui s'oppose aux approches classiques qui utilisent des modèles explicites de vélocité. Comparer l'approche proposée peut donc se faire à deux niveaux : premièrement avec les modèles classiques avec une densité de transition uniforme (c'està-dire avec un modèle similaire au nôtre, mais sans information floue), et deuxièmement avec des modèles utilisant une information de vélocité explicite (c'est-à-dire, utilisant une information similaire à la nôtre, mais dans un formalisme mathématique différent).

Ainsi, dans le but de fournir des résultats pertinents, nous comparons notre modèle au filtre SIR, au filtre particulaire auxiliaire (AUX-PF), au filtre particulaire à modèles multiples (MMPF) et au filtre particulaire auxiliaire à modèles multiples (AUX-MMPF), en utilisant un modèle dynamique à vélocité constante, et un modèle uniforme, dans le but de tester le comportement adaptatif que l'on peut attendre de notre approche, dans deux scénarios synthétiques. Dans le premier, des tests sur une séquence linéaire sont effectués (figure 4.1(a)), pour lesquels nous pourrions attendre une supérioté des modèles intégrant une information de vélocité sur les modèles uniformes. La seconde séquence (figure 4.1(b)) présente une dynamique quasi erratique, avec de fortes discontinuités, ce qui devrait favoriser les modèles uniformes. Dans la dernière partie de cette section, nous considérons également un vrai problème de suivi, montrant un hélicoptère téléguidé avec une dynamique complexe.

4.3.1.1 Scénario synthétique 1

Pour les modèles avec vélocité, nous avons fixé les paramètres dynamiques $[\mathbf{v}_t^1 \ \mathbf{v}_t^2]^T \sim \mathcal{N}(\mathbf{0}_{2\times 1}, 4^2\mathbf{I}_{2\times 2})$, avec $\mathbf{0} = [0 \ 0]^T$ et $\mathbf{I}_{2\times 2}$ la matrice identité 2×2 . Le coefficient de variance 4^2 a été choisi afin de fournir aux modèles comparatifs les meilleurs résultats possibles. Pour les modèles uniformes (c'est-à-dire nos approches et les modèles utilisant une simulation uniforme), nous fixons les paramètres d'horizons 2D à $[-40, 40] \times [-40, 40]$. Cela signifie que pour les modèles comparatifs utilisant une simulation uniforme, les paramètres de vélocité suivent la loi $\mathcal{U}[-40, 40]^2$ et non la marche aléatoire continue $\mathcal{N}(\mathbf{0}_{2\times 1}, 4^2\mathbf{I}_{2\times 2})$.

Les résultats illustrés sur la figure 4.16(b) ont été moyennés sur 100 réalisations du processus d'observation $\mathbf{y}_t = \mathbf{x}_t + \mathbf{w}_t$, avec $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}_{2\times 1}, 4^2\mathbf{I}_{2\times 2})$. Notons que les deux axes sont logarithmiques, dans un souci de clarté. Comme nous pouvions l'imaginer, dès que le nombre de particules est supérieur à 20, les modèles utilisant une information de vélocité sont supérieurs sur ce type de séquence linéaire. Nous pouvons également observer la supériorité globale des approches utilisant la dernière observation pour générer le nuage de particules (d'ailleurs cette remarque sera vraie pour tous les tests qui seront effectués dans cette section). Le FS-PF présente une erreur plus faible par rapport à toutes les autres méthodes qui n'utilisent pas la dernière observation (c'est-à-dire les filtres SIR et les MMPF), alors que le PFS-PF améliore de manière significative les résultats du FS-PF et surpasse les autres filtres lorsque le nombre de particules est faible. Nous pouvons également remarquer que la vitesse de convergence du PFS-PF est plus rapide que celle des autres méthodes. Lorsque N > 100, tous les filtres linéaires et nos approches fournissent des résultats comparables.

4.3.1.2 Scénario synthétique 2

Dans ce scénario, nous gardons les paramètres du scénario 1. Les résultats sont illustrés sur la figure 4.17(b). Dans ce cas, les modèles uniformes obtiennent des performances supérieures par rapport aux modèles linéaires, ce qui semble cohérent au vu des déplacements erratiques observés sur la figure 4.17(a). L'interprétation globale est identique à celle formulée dans le premier scénario. Cela montre, expérimentalement, le comportement adaptatif de nos approches, qui fournissent de meilleurs résultats dans les deux



FIG. 4.16 – (a) Premier scénario : séquence linéaire et (b) erreurs quadratiques moyennes obtenues par différents filtres selon le nombre de particules N pour ce scénario . Les deux axes sont logarithmiques.

scénarios. En particulier, comme nous l'avons vu dans la section 4.2 et comme nous le constatons maintenant, la force de notre approche avec fonction de proposition est de pouvoir exploiter les linéarités comme les situations erratiques, ce qui explique les bonnes performances obtenues.



FIG. 4.17 – (a) Second scénario : séquence avec une dynamique erratique et (b) erreurs quadratiques moyennes obtenues par différents filtres selon le nombre de particules N pour ce scénario. Les deux axes sont logarithmiques.

4.3.1.3 Séquence d'hélicoptère

Pour les modèles non uniformes, nous considérons des marches aléatoires continues pour les paramètres de vélocité, d'orientation et de dimension, avec une matrice de covariance diagonale pour le processus de bruit fixé à $\sigma_{\mathbf{v}_t^1}^2 = \sigma_{\mathbf{v}_t^2}^2 = 15^2$, $\sigma_{\mathbf{v}_t^3}^2 = 0.1$ et $\sigma_{\mathbf{v}_t^4}^2 = \sigma_{\mathbf{v}_t^5}^2 = 1$, respectivement. Pour les modèles uniformes, nous fixons les paramètres d'horizon de vélocité à $[-30, 30] \times [-30, 30]$ et de dimension de l'ellipse à $[-25, 25] \times [-25, 25]$. Les autres paramètres sont identiques à ceux mentionnés dans la section 4.3.1.1. Notons que les paramètres $\sigma_{\mathbf{v}_t^1}^2 = \sigma_{\mathbf{v}_t^2}^2$ ont été fixés de manière à ce que le filtre SIR fournisse les meilleurs résultats possibles à convergence.

La figure 4.18(b) présente les erreurs moyennes quadratiques, moyennées sur 100 lancés, de la position estimée de l'hélicoptère obtenue par les différents filtres. Nous pouvons premièrement noter que les modèles uniformes obtiennent de bons résultats lorsque le nombre de particules est faible mais stagnent assez rapidement (N > 500). D'un autre côté, les approches utilisant un modèle avec vélocité explicite présentent des erreurs plus faibles avec un fort nombre de particules (N > 2000). Encore, le FS-PF présente une supériorité sur les filtres SIR et MMPF. Le PFS-PF présente également une amélioration par rapport au FS-PF et aux autres modèles, surtout lorsque le nombre de particules est faible, ce qui montre la pertinence de la fonction d'importance. Là encore, la vitesse de convergence du PFS-PF est plus rapide que celle des autres méthodes.



FIG. 4.18 – (a) Trajectoire de l'hélicoptère et (b) erreurs quadratiques moyennes obtenues par différents filtres selon le nombre de particules N pour cette séquence. Les deux axes sont logarithmiques.

4.3.2 Temps de calcul et robustesse des modèles proposés

Nous passons maintenant aux tests concernant les temps de calcul et la robustesse des approches proposées. Nous utilisons une séquence synthétique et la séquence d'hélicotère. Pour les approches comparatives, seuls les modèles avec vélocité ont été étudiés dans cette section. En effet, ce sont ceux les plus fréquemment utilisés dans la littérature, et comme nous l'avons constaté dans la section 4.3.1.3, ils permettent d'obtenir de meilleurs résultats à convergence que ceux observés par les modèles uniformes.

4.3.2.1 Scénario synthétique 3

Nous utilisons ici la trajectoire synthétique 3. Elle présente la particularité d'être fortement linéaire à certains endroits, puis de brusquement changer de trajectoire ensuite (figure 4.19(a)). Nous considérons que ce type de comportement est typique dans les séquences de suivi. De plus, il favorise clairement les méthodes avec modèles à vélocité, et encore plus ceux qui utilisent des modèles multiples, une combinaison que nous considérons ici.

Le vecteur de bruit de position suit $[\mathbf{v}_t^1 \ \mathbf{v}_t^2]^T \sim \mathcal{N}(\mathbf{0}_{2\times 1}, 4^2 \times \mathbf{I}_{2\times 2})$, avec $\mathbf{0}_{2\times 1} = [0 \ 0]^T$ et $\mathbf{I}_{2\times 2}$ la matrice identité 2×2 . Pour les modèles proposés, les horizons de la vélocité sont fixés à $h_x = h_y = 40$. La forme des fonctions floues pour la vélocité est triangulaire. Pour les paramètres du modèle proposé utilisant une distribution d'importance, nous avons fait les mêmes choix, et pour les mêmes raisons, que ceux effectués dans la section 4.3.1.1, c'est-à-dire, $\gamma = 0.3$, m = 3, $\beta^1 = \beta^3 = 0.1$, $\beta^2 = 0.8$. La variance du bruit du processus de mesure est fixée à $\sigma_w^2 = 5$.



FIG. 4.19 – (a) Troisième scénario (séquence de suivi classique) et (b) erreurs quadratiques moyennes obtenues par différents filtres selon le nombre de particules N pour ce scénario. Les deux axes sont logarithmiques.

La figure 4.19(b) présente des erreurs moyennes quadratiques, moyennées sur 100 lancés, de la position estimée par le filtre SIR, le filtre AUX-PF, le MMPF, l'AUX-MMPF et nos approches FS-PF et PFS-PF. La figure 4.20 présente l'écart-type de l'estimateur obtenu par les différents filtres (équation 1.34). Cette métrique indique la disparité des erreurs obtenues, et permet d'attester de la stabilité stochastique de la méthode. Ainsi, une faible valeur d'ecart-type indique que les différences constatées entre plusieurs lancés de la méthode sont minces, ce qui est un comportement souhaitable d'une méthode par nature aléatoire.



FIG. 4.20 – Troisième scénario : écart-type des estimateurs donnés par les différents filtres, en fonction du nombre N de particules. Les deux axes sont logarithmiques.

Les résultats limités obtenus par le SIR et le MMPF sont dus aux discontinuités dans la dynamique (figures 4.19(a) et 4.21), qui sont mieux gérées lorsque la nouvelle observation est utilisée pour générer les particules. Pour les modèles proposés et l'AUX-MMPF, nous pouvons apprécier la décroissance rapide de l'erreur obtenue, ainsi que de l'écart-type de l'estimateur, en fonction du nombre de particules. Nous pouvons par exemple noter que les résultats obtenus par le FS-PF avec N = 20 particules sont meilleurs que ceux obtenus par le filtre SIR avec N = 2000.

La figure 4.22 présente les coûts des différents filtres en fonction de leurs erreurs moyennes quadratiques. Nous pouvons tout d'abord noter que le PFS-PF a un temps de calcul comparable à celui du AUX-MMPF alors que le temps de calcul du FS-PF est équivalent à celui du SIR. L'avantage du PFS-PF sur le FS-PF est clair en utilisant un faible nombre de particules $N \leq 100$, puisque pour la même erreur, le temps de calcul



FIG. 4.21 – Troisième scénario : (a) trajectoires estimées par les différents filtres sur un lancé avec N = 20 particules, et (b) zoom entre t = 15 et t = 23.

ainsi que la taille de la mémoire sont plus faibles. En utilisant plus de particules, il devient plus avantageux d'utiliser le FS-PF, puisque les deux méthodes convergent rapidement (figure 4.19).



FIG. 4.22 – Troisième scénario : coûts de calculs (en secondes, moyennés sur 100 lancés) en fonction des erreurs moyennes quadratiques obtenues par les différents filtres. Les deux axes sont logarithmiques.

4.3.2.2 Séquence d'hélicoptère

Fixer la variance du bruit $[\mathbf{v}_t^1 \ \mathbf{v}_t^2]^T$ du vecteur de position et de vélocité est une étape délicate dans la mise en œuvre d'un filtre de suivi. Cela nécessite une information *a priori* forte sur la dynamique de l'objet. Dans la pratique, c'est une information dont on ne dispose généralement pas, et qui est donc fixée de manière empirique. Elle est même souvent surestimée, afin de limiter les risques d'échecs du suivi. Dans la séquence d'hélicoptère étudiée dans la section 4.3.1.3, ces paramètres ont été fixés de manière à ce que le filtre de base, le SIR, fournisse les meilleurs résultats possibles à convergence. Malheureusement, ces valeurs ne sont pas forcément optimales pour les autres filtres, ce qui rend la comparaison entre les approches délicate. Dans cette section, nous choisissons une autre stratégie de sélection des paramètres. Le but ici étant de proposer une analyse dont les résultats sont associés aux meilleures performances en termes de temps de calculs, nous choisirons les paramètres qui correspondent aux meilleurs performances du filtre AUX-MMPF, selon un nombre de particules choisi. Nous fixons ce nombre de particules à N = 100, ce qui correspond à un traitement de 5 images par seconde, et semble donc bien adapté à ce

méthode choisie, 5 correspond à une moyenne). D'après la figure 4.23, l'erreur minimale atteinte par le filtre AUX-MMPF correspond aux variances $\sigma_{\mathbf{v}_t}^2 = \sigma_{\mathbf{v}_t}^2 = 4^2$ du vecteur de bruit de position $[\mathbf{v}_t^1 \ \mathbf{v}_t^2]^T$, valeurs que nous utiliserons donc dans les expériences faites dans cette section. Notons qu'à la vue de l'analyse faite dans la figure 4.23, le comportement des approches proposées reste stable quelles que soient les valeurs de variance considérées. Cela s'explique par le fait que ces paramètres ne sont utilisés que pour la transition du vecteur position \mathbf{x}_t (équation 4.1), les paramètres dynamiques étant estimés par le modèle flou.



FIG. 4.23 – Séquence d'hélicoptère : erreurs moyennes quadratiques obtenues par les différents filtres, pour un nombre de particules fixé N = 100. L'erreur minimale obtenue par le filtre AUX-MMPF correspond aux variances $\sigma_{\mathbf{v}_t^1}^2 = \sigma_{\mathbf{v}_t^2}^2 = 4^2$ du vecteur de bruit de position $[\mathbf{v}_t^1 \ \mathbf{v}_t^2]^T$.

Le vecteur de bruit de position suit $[\mathbf{v}_t^1 \ \mathbf{v}_t^2]^T \sim \mathcal{N}(\mathbf{0}_{2\times 1}, 4^2 \times \mathbf{I}_{2\times 2})$, avec $\mathbf{0}_{2\times 1} = [0 \ 0]^T$ et $\mathbf{I}_{2\times 2}$ la matrice identité 2×2 , le bruit d'orientation suit $\mathbf{v}_t^3 \sim \mathcal{N}(0, 0.1)$ et le bruit du vecteur de dimension suit $[\mathbf{v}_t^4 \ \mathbf{v}_t^5]^T \sim \mathcal{N}(\mathbf{0}_{2\times 1}, \mathbf{I}_{2\times 2})$. Les horizons de la vélocité sont fixés à $h_x = h_y = 60$ et les dimensions de l'ellipse à $h_w = h_h = 50$. La forme des fonctions floues pour la vélocité, l'orientation et la dimension sont triangulaires. Pour les paramètres du modèle proposé utilisant une distribution d'importance, les paramètres sont encore fixés à $\gamma = 0.3$, m = 3, $\beta^1 = \beta^3 = 0.1$ et $\beta^2 = 0.8$.

Nous nous focalisons maintenant sur la mise en œuvre de la distribution d'importance utilisée dans la méthode PFS-PF. Cela est très important puisque, comme nous l'avons vu dans la section 3.4, sa complexité est liée à la dimension du vecteur des paramètres et aux nombres de classes floues par paramètre, ce qui peut rendre le calcul de ce filtre très lourd si sa modélisation n'est pas soigneusement choisie. Ainsi, pour des raisons de temps de calcul, la distribution d'importance est utilisée uniquement pour le paramètre de vélocité. Nous avons également effectué une approximation de la vraisemblance utilisée dans le calcul de la fonction d'importance, en utilisant uniquement le premier terme de la distance entre histogrammes de la vraisemblance de région de l'équation 4.15, et sans utiliser de vraisemblance de contour (équation 4.10). Evidemment, cette approximation n'intervient que dans l'étape de simulation, la vraisemblance calculée dans la mise à jour des poids reste la même pour toutes les méthodes.

La figure 4.18(b) présente les erreurs moyennes quadratiques, moyennées sur 100 lancés, de la position de l'hélicoptère estimée par les différents filtres. Comme dans la section 4.3.1.3, les résultats affichés par les approches proposées sont meilleurs que ceux obtenus avec les autres filtres. Le cas particulier de l'AUX-PF s'explique par la faible valeur de variance utilisée dans le bruit de la dynamique.



FIG. 4.24 – (a) Trajectoire de l'hélicoptère et (b) erreurs quadratiques moyennes obtenues par différents filtres selon le nombre de particules N pour cette séquence. Les deux axes sont logarithmiques.

La figure 4.25 présente les écarts-types obtenus par les différents estimateurs. Nous pouvons apprécier la robustesse des approches proposées. Les fortes valeurs observées pour les autres approches s'expliquent par le fait que les filtres concernés perdent fréquemment l'hélicoptère, ce qui crée de fortes instabilités dans le calcul. Ainsi, les erreurs quadratiques moyennes obtenues restent fortement conditionnées par le succès ou l'échec du processus de suivi. Pour les approches proposées, la partition floue utilisée dans cette expérience décrit un large espace de recherche, ce qui limite les pertes d'objet puisque cela permet de gérer de fortes discontinuités de la dynamique.

La figure 4.26 montre les trajectoires estimées sur un lancé typique avec N = 50 particules. La figure 4.27 donne des images des suivis observés par les approches proposées, les modèles comparatifs et la vérité terrain avec N = 50 particules. Comme nous pouvons le constater, les filtres SIR, AUX-PF, MMPF et AUX-MMPF perdent l'hélicoptère aux environ des images 30 et 75. Cela est dû aux discontinuités lorsque le cameraman bouge, et au faible nombre de particules utilisé, alors que les modèles flous semblent plus robustes.



FIG. 4.25 – Séquence d'hélicoptère : écart-type des estimateurs donnés par les différents filtres, en fonction du nombre N de particules. Les deux axes sont logarithmiques.

La figure 4.28 présente les coûts de calcul des différents filtres, selon leurs erreurs quadratiques moyennes. Ici encore, les approches proposées obtiennent de meilleures performances avec des temps de calcul raisonnables. Le bénéfice du PFS-PF sur le FS-FS est net lorsque le nombre de particules utilisé est inférieur à 300. Cela signifie que la méthode PFS-PF est intéressante lorsqu'une cadence de plus d'une image par seconde doit être obtenu dans le processus de suivi.



FIG. 4.26 – Séquence d'hélicoptère : (a) trajectoires estimées par les différents filtres sur un lancé avec N = 50 particules, et (b) zoom entre t = 64 et t = 78.



FIG. 4.27 – Images des ellipses entourant l'hélicoptère estimées par les différents filtres pour N = 50 particules aux temps 12, 18, 23, 75 et 78. Vert : SIR, rouge : AUX-PF, violet : MMPF, jaune : AUX-MMPF, marron : FS-PF (modèle proposé sans fonction d'importance) et bleu : PFS-PF (modèle proposé avec fonction d'importance).



FIG. 4.28 – Séquence d'hélicoptère : coûts de calculs (en secondes) en fonction des erreurs moyennes quadratiques obtenues par les différents filtres. Les deux axes sont logarithmiques.

4.4 L'information de formes floues en pratique

Afin de montrer la généricité et la souplesse de l'approche proposée, nous présentons maintenant une application de suivi de formes, ici d'une bouche, utilisant une information spatiale floue. Les postures de lèvres sont donc représentées par des ensembles flous, telles qu'elles ont été définies dans la section 2.3. L'objectif de cette expérience est de démontrer l'efficacité du modèle de forme floue que nous avons défini dans un contexte de suivi. Pour rappel, la séquence ainsi que le modèle de vraisemblance ont été décrits dans la section 4.1.3.

Les fonctions floues ont été obtenues en considérant trois classes de sept échantillons chacune. Les tests ont été réalisés sur les 180 premières images. Nous avons utilisé un patron large avec des contraintes de forme en utilisant un nombre de points de contrôle P = 18, générant les probabilités de transition induites par les degrés d'intersection, obtenues dans la section 2.3.2.2, sans utiliser de seuil minimal d'intersection entre classes (équation 2.4). Les partitions floues des concepts d'orientation et de vélocité sont également de type triangulaire.

La distribution d'importance définie dans la section 3.4 n'a pas été utilisée dans cette section puisque l'intérêt principal de cette application est d'illustrer le potentiel de modélisation de l'approche. Ainsi, la difficulté soulevée ici concerne le suivi de formes, qui est lié à sa modélisation, plus que celui d'une dynamique complexe. Celle-ci ne nécessite donc pas un procédé de simulation évolué.

Le tableau 4.5 présente les erreurs moyennes de recouvrement obtenues en utilisant un nombre croissant de particules avec notre approche FS-PF et celle développée par Isard *et al.*. Le degré d'intersection correspond au ratio entre les points communs de la forme candidate CS et la vraie forme TS et le maximum des aires de ces deux objets, c'est-à-dire :

$$\frac{\sum_{p \in \Omega_I} \mathbb{1}_{CS}(p) \mathbb{1}_{TS}(p)}{\max\left[\sum_{p \in \Omega_I} \mathbb{1}_{CS}(p), \sum_{p \in \Omega_I} \mathbb{1}_{TS}(p)\right]}$$
(4.19)

avec Ω_I le domaine de l'image I et $\mathbb{1}_S(p)$ la fonction indicatrice qui vaut 1 si le point p appartient à l'object décrit par la forme S, 0 sinon.

La différence entre les résultats obtenus par ces deux méthodes peut s'expliquer de différentes façons. Tout d'abord, le modèle dynamique linéaire est assez restrictif, ce qui ne permet pas d'être robuste à des changements inopportuns de la dynamique. De plus,

Modèle	Différences d'aires (en %)					
	20	50	200	500	2000	5000
FS-PF	35.8	26.3	21.8	20.1	19.4	19.3
Isard <i>et al.</i>	44.7	39.3	24.2	23.4	21.9	20.4

TAB. 4.5 – Séquence de lèvres : erreurs des formes moyennes obtenues (en pourcentage) par les méthodes FS-PF et le modèle auto-régressif d'ordre 2 proposé par Isard *et al.,* en fonction du nombre N de particules.



FIG. 4.29 – Estimations des contours obtenus en utilisant N = 50 particules (a) par le FS-PF et (b) par la méthode d'Isard *et al.*.

un modèle auto-régressif du second ordre n'est pas vraiment adapté aux dynamiques périodiques, ce qui est le cas avec le mouvement de la bouche. Enfin, les vraisemblances utilisées par les deux modèles ne sont pas identiques, ce qui occasionne forcément une différence en pratique. Au contraire, le modèle flou proposé est plus souple, et permet de générer une plus grande variété de formes, ce qui est tout à fait adapté à une méthode d'approximation par simulation d'échantillons. Les résultats obtenus par les deux méthodes sont présentés sur la figure 4.29. La figure 4.30 illustre les erreurs observées par notre méthode sur la séquence entière avec N = 5000 particules.



FIG. 4.30 – Erreur de forme obtenue par le FS-PF avec N = 5000 particules sur 180 images. L'erreur est calculée comme l'aire des différences entre le résultat obtenu et la vérité terrain, en pourcentage.

L'estimation de la posture obtenue et les probabilités a posteriori des postures sont re-

présentées sur les figures 4.31(a) et 4.31(b). Sur la première figure, la valeur 1 correspond à une bouche fermée, 2 à une ouverte, et 3 à un sourire. Du fait que l'estimation est une valeur moyenne, les résultats obtenus sur la figure 4.31(a) sont graduels, et indiquent par conséquent des postures intermédiaires. Il est par ailleurs bon de noter ici qu'une information statistique de type moyenne n'est en fait pas très pertinente, puisqu'il s'agit d'indices de fonctions floues, qui peuvent ne pas être ordonnés, c'est pourquoi nous illustrons également les probabilités *a posteriori* obtenues des postures (figure 4.31(b)). Cependant, dans ce contexte assez simple, la forme bouche a une dynamique qui respecte l'ordre des indices des classes. La comparaison avec la vérité terrain montre que notre approche estime, en majorité, les bonnes postures. Bien qu'on puisse observer un léger accroissement des valeurs, les postures estimées sur les images ~ 150 et ~ 160 sont mises en échec pour une raison simple : cette période correspond à des passages où l'individu entrouve très légèrement la bouche (figures 4.32(e) et 4.32(f)). La vérité terrain n'intègre pas une information de degré d'ouverture de la bouche, ce qui explique ce résultat. Enfin, l'espérance de Monte-Carlo obtenue dans notre séquence de tests ainsi que l'estimation des indices des postures sont illustrées à divers temps sur la figure 4.32, en utilisant N = 5000particules.



FIG. 4.31 – (a) Estimation (espérance mathématique) de la posture obtenue par le FS-PF (la valeur 1 correspond à la posture de bouche *fermée*, 2 à *ouvert* et 3 à *sourire*) et (b) probabilités *a posteriori* des postures, avec N = 5000 particules et 180 images.

4.5 Conclusion

Nous avons réalisé dans ce chapitre une série d'expériences visant à montrer l'apport de l'intégration des relations spatiales floues et des informations de forme dans le filtre particulaire. Concernant les tests sur les relations spatiales floues, nous avons modélisé les paramètres de vélocité, d'orientation et d'échelle. Nous avons évalué notre modèle sans fonction de proposition (FS-PF), et avec fonction de proposition (PFS-PF), afin de mieux mesurer l'apport de celle-ci dans le suivi d'objets.

La première série d'expériences étudie l'influence des paramètres dans nos approches floues. Tout d'abord, une granulométrie de 3 fonctions floues par dimension pour représenter la vélocité conduit à de bonnes performances, et possède également l'avantage d'être simple à mettre en place. En outre, utiliser un petit nombre de valeurs pour représenter un concept flou permet de mieux mettre en avant le modèle de sauts dont les paramètres sont définis par les degrés d'intersection entre les ensembles flous. La forme



FIG. 4.32 – Estimations des contours et des postures obtenus par le FS-PF aux instants (a) t = 11, (b) t = 66, (c) t = 140, (d) t = 146, (e) t = 151 et (f) t = 159.

des fonctions d'appartenance a également son importance quant à la réussite du suivi. Et ici encore, le choix d'une fonction triangulaire par rapport à une forme rectangulaire ou trapèzoïdale semble judicieux, puisque cela permet également de définir des transitions souples entre les états, tout en localisant de manière précise l'objet. Comme nous pouvions l'imaginer, la longueur du support des fonctions floues a également son importance. Cela a été testé avec le paramètre de vélocité. Dans nos tests, la plage de valeurs donnant des résultats acceptables se trouve assez large, ce qui permet de facilement fixer cette valeur, avec une connaissance *a priori* faible sur la séquence. Au sujet de la fonction de proposition, nous avons fait varier la longueur de la trajectoire afin de mesurer son impact sur la qualité de l'estimation face à deux scénarios : l'un très linéaire, et l'autre totalement erratique. Dans ces deux expériences, il semble qu'utiliser un historique long permet d'améliorer la qualité du suivi.

Nous avons ensuite étudié le comportement de nos approches dans des séquences dont les dynamiques sont linéaires ou erratiques, cela dans le but de montrer le caractère adaptatif du modèle flou souple défini, qui permet d'exploiter les linéarités comme les changements brusques de trajectoire, notamment grâce à la fonction de proposition. Les expériences ont été réalisées en utilisant deux séquences synthétiques, et une séquence réelle, montrant un hélicoptère téléguidé, de dynamique changeante. Ces tests ont montré une supérioté de nos approches en termes de performances face aux modèles de suivi classiques SIR et AUX-PF, et aux modèles à sauts dynamiques MMPF, AUX-MMPF, qui sont des modèles de référence dans la communauté du suivi d'objets. Le bénéfice de la fonction de proposition dans l'approche PFS-PF est net, surtout lorsque le nombre de particules est faible, ce qui témoigne de la pertinence d'utiliser une information de trajectoire, ainsi que la dernière observation, dans le processus de simulation des particules. Les méthodes FS-PF et PFS-PF convergent assez vite lorsque le nombre de particules augmente, ce qui permet, dans une application pratique, de limiter le nombre de particules tout en observant des performances de suivi satisfaisantes. Enfin, nous avons testé les performances en temps de calcul sur une séquence synthétique et la séquence d'hélicoptère. Les résultats montrent que l'utilisation de la méthode PFS-PF est plus avantageuse que la méthode FS-PF lorsque nous souhaitons garder un temps de calcul raisonnable (par exemple, au-dessus de 1 image par seconde). Les deux approches proposées (PFS-PF et FS-PF) restent toujours avantageuses face aux autres méthodes comparées, les filtres SIR, AUX-PF, MMPF et AUX-MMPF. Dans le cadre du suivi de l'hélicoptère, et dans le but de vérifier si les différences observées sont significatives, nous avons par ailleurs effectué le test de Welch (Welch's t test en anglais) [Welch, 1947], qui est une adaptation du test de Student dans le cas où les deux échantillons peuvent avoir des variances différentes. Avec un risque de 0.05, cela indique que la différence d'erreurs entre les deux méthodes proposées n'est plus significative lorsque le nombre de particules est supérieur à 2000, alors que la comparaison avec les autres méthodes reste toujours significative.

Nous avons également proposé une application de suivi de formes non rigides, des bouches, représentées par des fonctions floues. Celles-ci ont été automatiquement apprises via une base de données disponible librement sur Internet. Nous avons considéré trois ensembles flous, représentant des postures : *fermée, ouvert* et *sourire*. La dynamique entre les postures de bouches a également été automatiquement générée grâce à notre formalisme flou. Les résultats obtenus ont été comparés à la méthode développée dans [Isard et Blake, 1998a], et semblent convaincants, puisque, notamment, l'approche proposée donne une erreur de formes de 26.3% avec 50 particules, alors que la méthode d'Isard *et al.*donne 39.3% d'erreur.

Chapitre 5

Intégration de contraintes spatiales floues pour le suivi multi-objets

Sommaire

5.1	Modé	Aodéliser des contraintes spatiales floues entre objets		
	5.1.1	Modélisation des contraintes spatiales floues		
	5.1.2	Intégration des contraintes spatiales floues dans un contexte pro-		
		babiliste		
5.2	Filtrag	trage particulaire pour le suivi multi-objets		
	5.2.1	Modélisation d'un filtre particulaire multi-objets		
	5.2.2	Problème de la dimension du vecteur d'état		
5.3	L'écha	antillonnage partitionné		
	5.3.1	Rééchantillonnage pondéré		
	5.3.2	Echantillonnage partitionné		
	5.3.3	Discussion		
5.4	Echan	tillonnage partitionné à rang 141		
5.5	.5 Expériences \dots 146			
	5.5.1	Description des séquences		
		5.5.1.1 Suivi de piétons		
		5.5.1.2 Suivi de fourmis		
		5.5.1.3 Suivi de main		
	5.5.2	Suivi de piétons		
	5.5.3	Suivi de fourmis		
	5.5.4	Suivi de main		
5.6	Concl	Conclusion 161		
0.0				

Traiter plusieurs objets dans un filtre particulaire soulève plusieurs problèmes. Premièrement, il est souvent nécessaire de modéliser les interactions possibles entre objets. Cela est essentiel pour estimer conjointement tous les objets. Dans la plupart des approches, ce point est résolu en utilisant des hypothèses spécifiques (voir par exemple [Khan et al., 2005]), directement liées à l'application, ce qui peut rendre la généralisation du modèle ou son adaptation à d'autres problématiques difficile. Par ailleurs, les contraintes spatiales floues ont suscité un grand intérêt dans nombre de domaines, tels que le partitionnement de données (*data clustering* en anglais) [Pham, 2002], la segmentation de cerveaux dans des images IRM 3D [Colliot et al., 2006] ou encore le raisonnement par graphe sur des attributs flous [Fouquier et al., 2007]. Ainsi, dans ce chapitre, nous allons tout d'abord proposer de représenter ces interactions en utilisant un formalisme flou, ce qui permet de facilement prendre en compte des relations spatiales entre objets potentiellement complexes, dans un cadre unifié et formel.

L'adaptation des filtres particulaires au suivi multi-objets à été étudié un grand nombre de fois dans la littérature, de différentes manières. Parmi celles-ci, les auteurs dans [Doucet et al., 2002] proposent un système markovien à sauts (JMS en anglais, pour *Jump Markov System*) pour modéliser le nombre d'objets, les hypothèses d'associations entre les mesures et les objets et les états individuels. Dans [Schulz et al., 2001], les auteurs utilisent un filtre particulaire et modélisent des interactions entre objets et mesures en utilisant un filtre joint à association de données probabilistes (JPDAF en anglais, pour *Joint Probabilistic Data Association Filter*). Dans [Hue et al., 2002], la distribution des hypothèses d'associations est calculée en utilisant un échantillonneur de Gibbs.

D'un autre côté, comme signalé dans [MacKay, 1998; Robert et Casella, 2005] ainsi que dans les sections 1.1.2.3 et 1.3.1.4, l'échantillonnage par importance, procédure utilisée en particulier dans les filtres particulaires, souffre d'un problème lié à la dimension de l'espace d'état. Cela signifie que le filtre particulaire requiert un nombre de particules qui augmente de manière exponentielle avec le nombre d'objets, rendant l'utilisation en pratique du filtre particulaire pour le suivi multi-objets difficile dès que le nombre d'objets est supérieur à trois. Ainsi, les auteurs dans [MacCormick et Blake, 2000; MacCormick et Isard, 2000] proposent un filtre particulaire qui atténue ce surcoût en utilisant une stratégie d'échantillonnage partitionné, et en considérant un principe d'exclusion (c'està-dire en spécifiant qu'une mesure ne peut être associée qu'à un objet au plus). D'autres approches visant à réduire le coût lié à la dimension de l'état seront citées dans la section 5.2.

La stratégie d'échantillonnage partitionné est mise en œuvre en partitionnant l'espace d'état, en considérant typiquement un élément de la partition par objet, selon un ordre spécifique de traitement des objets, que nous appelons scénario, et en sélectionnant les particules jugées pertinentes par une procédure de rééchantillonnage pondéré. L'ordre considéré importe puisqu'il peut mener à des comportements indésirables du filtre, comme l'échec du suivi, par exemple si le premier objet considéré est caché par les autres. En fait, comme cela a été remarqué dans [Smith et Gatica-Perez, 2004] et comme nous le verrons plus tard dans ce chapitre, utiliser un ordre spécifique pour estimer l'état joint peut avoir plusieurs effets nuisibles, qui peuvent être expliqués par un phénomène d'appauvrissement du nuage de particules. Ainsi dans [Smith et Gatica-Perez, 2004], la loi de filtrage jointe est représentée par une loi de mélange, dans laquelle chaque composante désigne un ordre spécifique de traitement et est estimée en utilisant une technique d'échantillonnage partitionné. Cette idée a également été mise à profit dans [Duffner et al., 2009] pour fusionner plusieurs caractéristiques d'un seul objet. Cependant, le nombre de particules allouées pour chaque composante est fixé, ce qui peut dégrader les performances du filtre quand les ordres choisis ne sont pas pertinents [Smith et Gatica-Perez, 2004].

Dans ce chapitre, nous proposons d'intégrer des contraintes spatiales floues dans un filtre particulaire multi-objets et d'estimer de manière jointe les objets et l'ordre dans lequel les objets seront traités. Nous appelons cette dernière procédure échantillonnage partitionné à rang. Celle-ci permet de considérer l'ensemble entier des permutations pos-

sibles et d'élaguer implicitement les scénarios inadéquats.

5.1 Modéliser des contraintes spatiales floues entre objets

5.1.1 Modélisation des contraintes spatiales floues

Dans cette section, nous proposons de modéliser explicitement les interactions entre objets via des relations spatiales floues définies sur un ou plusieurs objets indiquant avec quel degré une relation est satisfaite. Celles-ci seront considérées comme des contraintes que les objets devront satisfaire, avec un degré non nul, c'est pourquoi nous les nommons contraintes spatiales floues. Comme dans le chapitre 2, où nous avons modélisé des relations entre les positions d'un objet à des instants successifs, chaque relation est considérée comme une variable linguistique, prenant un petit nombre de valeurs linguistiques [Zadeh, 1975]. La granulométrie de cette représentation peut être définie en fonction de l'application. La sémantique de chacune des valeurs linguistiques est définie par un ensemble flou sur le domaine de la variable. Les contraintes spatiales floues peuvent être définies par un opérateur flou unaire, tel que le concept de dimension d'un objet (dont les valeurs pourraient être *petit, moyen, grand,* ...); par des opérateurs binaires, tel que le concept d'orientation relative (dont les valeurs pourraient être est à droite de, est à gauche de, ...) ; par des opérateurs ternaires, tel que le concept de disposition locale (dont les valeurs pourraient être *est le premier de, est au milieu de, est le dernier de*); et plus généralement par des opérateurs *n*-aires. Dans le cadre de ce travail, nous nous focaliserons sur les opérateurs binaires, ternaires et quaternaires, en considérant les concepts d'intersection, de distance, d'angle et d'alignement. Les contraintes spatiales floues peuvent être fixées tout au long du processus de suivi (la configuration spatiale des objets est fixée, bien que connue imprécisément), peuvent évoluer dans le temps (les concepts spatiaux flous changent graduellement leurs valeurs), ou peuvent encore être définies par l'espace et le temps (en considérant des contraintes spatio-temporelles imprécises). Ici nous ne considérerons que des relations spatiales indépendantes du temps bien que ces deux derniers types de contraintes puissent certainement être envisagés.

L'idée consiste à définir une valeur linguistique qui traduit la satisfaction des contraintes spatiales pour une configuration d'objets donnée. Cette valeur linguistique est définie par un ensemble flou, qui évalue donc le degré de satisfaction des contraintes spatiales pour cette configuration d'objets. Contrairement à la modélisation des relations spatiales étudiées dans le chapitre 3, nous considérons dans ce chapitre non pas une relation spatiale d'un objet entre deux instants successifs, mais les relations spatiales entre les objets modélisés de la scène, à un instant donné. Les contraintes spatiales définies dans ce chapitre sont de nature complexe, et ne peuvent généralement pas être évaluées de manière analytique, comme c'était le cas avec les relations spatiales définies dans le chapitre 3. Ainsi, à défaut de guider les particules pendant l'étape de prédiction, elles serviront ici à évaluer le degré de possibilité d'une configuration spatiale des objets.

Comme nous le verrons dans la section 5.3, notre méthode d'estimation de la loi jointe sur tous les objets est séquentielle, et nécessite donc d'évaluer un nouvel objet, par rapport aux précédents. Ainsi, l'ensemble flou des contraintes spatiales est défini par l'ensemble des objets qui ont déjà été traités. En évaluant la valeur d'appartenance obtenue par l'objet à traiter, il définit donc le degré de satisfaction des contraintes spatiales du nouvel objet par rapport à ceux déjà validés. Afin d'intégrer plusieurs contraintes spatiales, l'ensemble flou est défini par une fusion floue de chacun des degrés de satisfaction

d'une contrainte spatiale. Soit \mathbf{x}_t^* l'état hypothétique d'un objet et $\tilde{\mathbf{x}}_t = (\tilde{\mathbf{x}}_t^1, \dots, \tilde{\mathbf{x}}_t^L)$ le vecteur d'état de *L* objets déjà traités au temps *t*. Nous définissons tout d'abord une première fonction d'appartenance $\nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*) \in [0, 1]$ qui décrit avec quel degré la configuration d'un objet \mathbf{x}_t^* satisfait les contraintes spatiales supposées connues imposées par $\tilde{\mathbf{x}}_t$. Notons *K* le nombre de contraintes spatiales que nous considérons. Soit $\nu_{\tilde{\mathbf{x}}_t}$ tel que :

$$\nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*) = \frac{\overset{K}{\Xi}}{\underset{k=1}{\overset{*}{\Xi}}} \overset{*}{\nu}_{\tilde{\mathbf{x}}_t}^k(\mathbf{x}_t^*)$$
(5.1)

avec Ξ un opérateur de fusion, par exemple une t-norme (conjonction floue) [Dubois et Prade, 1980], et $\nu_{\tilde{\mathbf{x}}_t}^k \in [0, 1]$ la fonction d'appartenance de la *k*-ième contrainte spatiale, qui désigne donc avec quel degré l'objet \mathbf{x}_t^* satisfait la contrainte spatiale *k* imposée par les objets $\tilde{\mathbf{x}}_t$. Par exemple, en considérant une relation floue binaire, ${}^B\!\nu_{\tilde{\mathbf{x}}_t}^k$ est définie par :

$${}^{B}\!\nu_{\tilde{\mathbf{x}}_{t}}^{k}(\mathbf{x}_{t}^{*}) = \underset{l=1}{\overset{L}{\psi}}{}^{B}\!\nu_{\tilde{\mathbf{x}}_{t}^{l}}^{k}(\mathbf{x}_{t}^{*})$$
(5.2)

avec ψ un opérateur de fusion, par exemple une t-norme, et ${}^{B}\nu_{\tilde{\mathbf{x}}_{t}^{k}}^{k} \in [0,1]$ la fonction d'appartenance définissant la k-ième contrainte spatiale entre l'objet courant et l'objet l. Par exemple, si \mathbf{x}_{t} correspond aux coordonnées spatiales d'un objet, alors la valeur à *moyenne distance* peut être définie par la fonction floue trapézoïdale :

$${}^{B}\!\nu_{\tilde{\mathbf{x}}_{t}^{l}}^{k}(\mathbf{x}_{t}^{*}) = \begin{cases} \frac{\|\mathbf{x}_{t}^{*} - \tilde{\mathbf{x}}_{t}^{l}\|_{2} - a}{b - a} & \text{si } \|\mathbf{x}_{t}^{*} - \tilde{\mathbf{x}}_{t}^{l}\|_{2} \in]a; b[\\ 1 & \text{si } \|\mathbf{x}_{t}^{*} - \tilde{\mathbf{x}}_{t}^{l}\|_{2} \in [b, c] \\ 1 - \frac{\|\mathbf{x}_{t}^{*} - \tilde{\mathbf{x}}_{t}^{l}\|_{2} - c}{d - c} & \text{si } \|\mathbf{x}_{t}^{*} - \tilde{\mathbf{x}}_{t}^{l}\|_{2} \in]c; d[\\ 0 & \text{sinon} \end{cases}$$

avec a, b, c, d les paramètres d'une fonction trapézoïdale floue de support [a, d] et de noyau [b, c]. La forme de la fonction ${}^{B}\nu_{\tilde{\mathbf{x}}_{t}^{l}}^{k}$ peut être fixée par l'application. La figure 5.1 présente un exemple de contraintes spatiales d'orientation entre les doigts d'une main, en utilisant une forme d'appartenace rectangulaire, donc une appartenance binaire. Le *majeur* correspond à l'état $\mathbf{x}_{t}^{*} = \mathbf{x}_{t}^{majeur}$ dont on souhaite contraindre la position centrale, et les doigts *index*, *annulaire* et *auriculaire* forment le vecteur d'états concaténés $\tilde{\mathbf{x}}_{t} = (\tilde{\mathbf{x}}_{t}^{index}, \tilde{\mathbf{x}}_{t}^{annulaire}, \tilde{\mathbf{x}}_{t}^{auriculaire})$, qui ont déjà été traités. Ainsi, la contrainte spatiale binaire associée à l'*index* ${}^{B}\nu_{\tilde{\mathbf{x}}_{t}^{index}}(\mathbf{x}_{t}^{majeur})$ traduit la valeur d'orientation *est à droite de* (figure 5.1(a)), tandis que les contraintes spatiales associées à l'*annulaire* et l'*auriculaire* traduisent la valeur *est à gauche de* (figures 5.1(b) et 5.1(c)). En utilisant un opérateur de fusion entre les valeurs floues d'orientation ψ (équation 5.2) de type t-norme min, nous obtenons le résultat global contraint par l'orientation (figure 5.1(d)).

D'une manière similaire, en considérant une contrainte ternaire, ${}^{T}\!\nu_{\tilde{\mathbf{x}}_{t}}^{k}$ est définie par :

$${}^{T}\nu_{\tilde{\mathbf{x}}_{t}}^{k}(\mathbf{x}_{t}^{*}) = \underset{l_{1}=1}{\overset{L}{\psi}} \overset{L}{\overset{L}{\psi}} {}^{T}\nu_{\tilde{\mathbf{x}}_{t}^{l_{1}},\tilde{\mathbf{x}}_{t}^{l_{2}}}^{k}(\mathbf{x}_{t}^{*})$$
(5.3)

avec ${}^{T}\!\nu_{\tilde{\mathbf{x}}_{t}^{l_{1}},\tilde{\mathbf{x}}_{t}^{l_{2}}}^{k} \in [0,1]$ la fonction d'appartenance de la *k*-ième contrainte spatiale entre l'objet courant et les objets l_{1} et l_{2} .



FIG. 5.1 – Contraintes spatiales floues du concept d'orientation dans le cas de la main, par rapport au doigt *majeur* : (a) valeur *est à gauche de l'index*, (b) valeur *est à droite de l'annulaire*, (c) valeur *est à droite de l'auriculaire* et (d) résultat global contraint en utilisant un opérateur de fusion $\psi = \min$. Le noir indique que la contrainte est satisfaite. Les objets déjà traités (*index, annulaire* et *auriculaire*) sont représentés en rouge.

Finalement, la fonction d'appartenance finale $\mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)$ décrivant les contraintes spatiales floues imposées par un groupe d'objets $\tilde{\mathbf{x}}_t$ est définie par la fonction $\nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)$, en y ajoutant un paramètre $\gamma \in \mathbb{R}^+$ fixé servant à contrôler la forme de la fonction μ :

$$\mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*) = \nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)^{\gamma} \tag{5.4}$$

avec $\nu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)$ la fonction d'appartenance des contraintes spatiales définies dans l'équation 5.1. Des exemples de contraintes binaires, ternaires et quaternaires modélisées par ${}^B\!\nu_{\tilde{\mathbf{x}}_t^l}^k, {}^T\!\nu_{\tilde{\mathbf{x}}_t^{l_1}, \tilde{\mathbf{x}}_t^{l_2}}^k$ et ${}^Q\!\nu_{\tilde{\mathbf{x}}_t^{l_1}, \tilde{\mathbf{x}}_t^{l_2}, \tilde{\mathbf{x}}_t^{l_3}}^k$ seront proposées dans la section 5.5. La prochaine étape consiste à intégrer la fonction finale des contraintes spatiales $\mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)$ dans un contexte probabiliste. Cela est l'objet de la section suivante.

5.1.2 Intégration des contraintes spatiales floues dans un contexte probabiliste

L'idée de l'intégration des contraintes spatiales floues entre objets dans un contexte probabiliste est assez similaire à celle exposée dans le chapitre 3 d'intégration des informations spatiales floues dans le filtre particulaire pour le suivi mono-objet. En particulier, nous allons utiliser le concept d'événements flous, en intégrant les contraintes spatiales floues $\mu_{\tilde{\mathbf{x}}_t}$ imposées par un groupe d'objets déjà traités $\tilde{\mathbf{x}}_t$ pour guider la transition d'un nouvel objet \mathbf{x}_t^* . Nous considérons ainsi un événement flou de contraintes dont la fonction d'appartenance s'écrit $\mu_{\tilde{\mathbf{x}}_t} \in [0, 1]$. De plus, la transition de ce nouvel objet doit également utiliser son état précédent, \mathbf{x}_{t-1}^* , afin d'utiliser la dynamique de celui-ci. La dynamique peut être décrite par un modèle de transition classique, ou le modèle flou proposé dans les chapitres 3 et 4. L'objectif consiste donc à définir une loi de transition qui dépende des contraintes imposées par les objets déjà traités $\tilde{\mathbf{x}}_t$ et de l'état précédent \mathbf{x}_{t-1}^* , soit une loi de transition $\mathbb{P}(\mathbf{x}_t^* \in d\mathbf{x}_t^* | \mathbf{x}_{t-1}^*, \tilde{\mathbf{x}}_t)$, avec $d\mathbf{x}_t^*$ un événement flou de support infinitésimal centré en \mathbf{x}_t^* et défini par une fonction d'appartenance $\mu_{d\mathbf{x}_t^*}$. En réutilisant la loi conditionnelle d'un événement flou sachant un autre événement flou définie dans l'équation 3.3, la loi de transition $\mathbb{P}(\mathbf{x}_t^* \in d\mathbf{x}_t^* | \mathbf{x}_{t-1}^*, \tilde{\mathbf{x}}_t)$ s'écrit :

$$\mathbb{P}(\mathbf{x}_{t}^{*} \in d\mathbf{x}_{t}^{*} | \mathbf{x}_{t-1}^{*}, \tilde{\mathbf{x}}_{t}) = \frac{\int_{\mathcal{X}^{*}} \mu_{d\mathbf{x}_{t}^{*}}(\mathbf{x}) \, \mu_{\tilde{\mathbf{x}}_{t}}(\mathbf{x}) \, p(\mathbf{x} | \mathbf{x}_{t-1}^{*}) d\mathbf{x}}{\int_{\mathcal{X}^{*}} \mu_{\tilde{\mathbf{x}}_{t}}(\mathbf{x}) p(\mathbf{x} | \mathbf{x}_{t-1}^{*}) d\mathbf{x}} \\
= D(\tilde{\mathbf{x}}_{t}, \mathbf{x}_{t-1}^{*}) \, d\mathbf{x}_{t}^{*} \mu_{\tilde{\mathbf{x}}_{t}}(\mathbf{x}_{t}^{*}) p(\mathbf{x}_{t}^{*} | \mathbf{x}_{t-1}^{*}) \tag{5.5}$$

avec \mathcal{X}^* l'espace de définition de \mathbf{x}^* et $\mu_{\tilde{\mathbf{x}}_t}$ les contraintes spatiales floues imposées par $\tilde{\mathbf{x}}_t$ définies dans l'équation 5.4. La seconde ligne de l'équation 5.5 est obtenue en utilisant le fait que l'ensemble flou $d\mathbf{x}_t^*$ est de support infinitésimal centré sur \mathbf{x}_t^* . Le terme de normalisation C peut être estimé par un algorithme d'acceptation/rejet (section 1.1.2.2), en utilisant un nombre d'échantillon peu élevé. Par ailleurs, dans nos tests, nous avons remarqué que considérer ce terme comme constant ne dégradait que peu les résultats, avec le bénéfice d'un coût en termes de temps de calculs moins élevé.

Dans un cadre général, il est difficile de définir $\mu_{\tilde{\mathbf{x}}_t}$ de manière analytique, ce qui rend la simulation selon l'équation $\mathbb{P}(\mathbf{x}_t^* \in d\mathbf{x}_t^* | \mathbf{x}_{t-1}^*, \tilde{\mathbf{x}}_t)$ impossible. Le terme $\mu_{\tilde{\mathbf{x}}_t}(\mathbf{x}_t^*)$ est ainsi considéré dans la phase de vraisemblance, ce qui laisse uniquement le terme $p(\mathbf{x}_t^* | \mathbf{x}_{t-1}^*)$ et rend la simulation des particules possible. Nous reviendrons sur ce point dans la section 5.3. Une extension de la loi de transition 5.5, qui n'a pas été abordée durant nos travaux, serait de considérer des contraintes pour lesquelles une expression analytique de la fonction $\mu_{\tilde{\mathbf{x}}_t}$ est connue. Dans ce cas, la simulation directe selon l'équation 5.5 resterait difficile (à part dans des cas très simples), mais présenterait tout de même la possibilité de générer les particules selon les contraintes $\mu_{\tilde{\mathbf{x}}_t}$, et ensuite, pendant la mise à jour des poids, de prendre en compte la transition $p(\mathbf{x}_t^* | \mathbf{x}_{t-1}^*)$. Etudier ce cas particulier constituerait peut être une perspective intéressante à ce travail.

5.2 Filtrage particulaire pour le suivi multi-objets

5.2.1 Modélisation d'un filtre particulaire multi-objets

Nous considérons fixé le nombre M d'objets à suivre. Soit $\mathbf{x}_t = (\mathbf{x}_t^1, \dots, \mathbf{x}_t^M)$ le vecteur aléatoire caché, avec $\mathbf{x}_t^i \in \mathcal{X}^i$ le *i*-ème objet. La littérature sur le sujet fourmille de travaux, dont il est difficile d'opérer une classification tant les approches peuvent être différentes. La première approche est certainement celle de [Avitzour, 1995], et consiste simplement à utiliser le filtre SIR dans le cas où \mathbf{x}_t est une concaténation de plusieurs objets, soit $\mathbf{x}_t = (\mathbf{x}_t^1, \dots, \mathbf{x}_t^M)$, sans opérer plus de changements.

Pourtant, un problème inhérent au suivi multi-objets ne permet pas de calculer la vraisemblance directement. Qui dit plusieurs objets, dit également plusieurs mesures, plusieurs observations. Le problème est donc de savoir à quelle(s) mesure(s) un objet est lié, afin de correctement calculer la vraisemblance de celui-ci. Si $\mathbf{y}_t = (\mathbf{y}_t^1, \dots, \mathbf{y}_t^O)$ désigne le vecteur d'observations, alors il convient de trouver la bonne correspondance qui lie les objets $\{\mathbf{x}_t^i\}_{i=1}^M$ aux observations $\{\mathbf{y}_t^j\}_{j=1}^O$. Pour ce faire, il faut déjà poser des hypothèses : un objet peut-il être lié à plusieurs mesures ? au contraire, un objet peut-il n'être lié à aucune mesure ? une mesure peut-elle être une fausse alarme ? une mesure peut-elle provenir de plusieurs objets ? etc. Toutes les réponses possibles à ces questions

donnent des modèles probabilistes d'association des données (observations) aux cibles (que nous appelons objets en traitement d'images) différents. Par exemple, les auteurs dans [Hue et al., 2002] considèrent qu'une mesure peut provenir d'un objet ou être une fausse alarme et qu'un objet peut être à l'origine d'aucune ou de plusieurs mesures. Les auteurs dans [Schulz et al., 2001] considèrent également qu'une mesure peut provenir d'un objet ou être une fausse alarme mais qu'un objet peut être à l'origine d'aucune ou d'au plus une mesure. Des méthodes un peu plus exotiques ont également vu le jour, telles que celle de [Yang et al., 2007], qui proposent de représenter le problème d'association des données aux objets par un formalisme issu de la théorie des jeux, dans le but de trouver l'équilibre de Nash qui indique qu'une configuration globale satisfaisante à été trouvée. D'autres filtres multi-objets classiques ont également été proposés, tels que ceux décrits dans [Doucet et al., 2002; Kreucher et al., 2004, 2005; Okuma et al., 2004; Vermaak et al., 2005a]...

Cependant, en vision, il n'est pas rare de relâcher un peu le problème d'association, et de ne considérer que l'hypothèse indiquant qu'une mesure peut provenir d'un objet ou être une fausse alarme. C'est ce qui a notamment été fait dans [MacCormick et Blake, 2000; Isard et MacCormick, 2001], et cela permet de gérer le problème lié à l'incertitude sur l'origine des mesures en ne modélisant qu'un problème de visibilité de l'objet, ce qui est plus simple, et peut par exemple être fait en modélisant explicitement un vecteur de visibilité et en l'estimant par l'algorithme du filtre particulaire [MacCormick et Blake, 2000; Pérez et Vermaak, 2005]. Cela implique que le processus de mesure dépende de la visibilité des objets. Bien qu'il soit possible de considérer une telle modélisation dans notre approche, nous avons choisi de gérer la visibilité de l'objet de manière implicite, comme cela a auparavant été effectué dans [Isard et MacCormick, 2001], dans lequel la visibilité est déterminée en estimant la position de l'objet dans la scène 3*D*.

Dans notre approche, la visibilité est implicitement gérée dans la vraisemblance en considérant qu'un objet placé tôt dans un scénario est toujours plus visible qu'un objet placé après, comme cela a déjà été considéré dans [Smith et Gatica-Perez, 2004]. Notons toutefois, comme cela a été signalé dans [Hue, 2003], qu'utiliser une hypothèse qui stipule qu'une mesure ne peut provenir qu'au plus d'un objet peut se trouver erronée lorsque deux objets sont proches, puisque cela dépend de la résolution du capteur, et surtout, cela dépend des observations qui ont été extraites de l'image, qui peuvent être liées à un ensemble de pixels.

5.2.2 Problème de la dimension du vecteur d'état

L'algorithme du filtre particulaire utilise, par essence, une procédure d'échantillonnage par importance lors de la simulation selon la densité de probabilité $q(\mathbf{x}_t | \mathbf{x}_{0:t-1}, \mathbf{y}_t)$ (section 1.2), qu'elle corresponde simplement à la loi de transition *a priori* $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ ou qu'elle soit plus évoluée. En pratique, cela signifie qu'une augmentation de la dimension de \mathbf{x}_t induit systématiquement une augmentation de la variance des poids des particules, ce qui peut mener à un phénomène d'appauvrissement du nuage de particules (c'est-à-dire que peu de particules différentes représentent l'objet). Ainsi, proposer un filtre multi-objets rejoint la problématique des filtres particulaires opérant en grande dimension [Snyder et al., 2008]. De tels filtres ont notamment été proposés dans [Bray et al., 2007; Vaswani, 2008], ou encore le filtre particulaire *Rao-Blackwellised* bien connu dans [Doucet et al., 2000a]. Dans le cadre du suivi multi-objets, où \mathbf{x}_t^i et \mathbf{x}_t^j sont deux états (ou objets) évoluant dans le même espace d'état \mathcal{X}^* , les auteurs dans [MacCormick et Isard, 2000] montrent que N^2 particules sont nécessaires pour obtenir le même niveau de performance qu'en traitant un seul objet avec N particules. Pour le voir, nous reprenons ici sommairement les arguments donnés par [MacCormick et Isard, 2000; MacCormick, 1999], en définissant tout d'abord le concept de diagnostic de survie, qui, pour un nuage de particules $\{\mathbf{x}_t^{(n)}, w_t^{(n)}\}_{n=1}^N$ donné, est calculé selon :

$$\mathcal{D} = \left[\sum_{n=1}^{N} \left(w_t^{(n)}\right)^2\right]^{-1} \tag{5.6}$$

Le diagnostic de survie correspond au nombre de particules qui devraient survivre à l'étape de rééchantillonnage. Dans le chapitre 1, nous avons appelé cette quantité le nombre efficace de particules (section 1.28). Les auteurs définissent ensuite le taux de survie, qui est calculé à l'aide d'une densité de probabilité *a priori* $\tilde{p}(\mathbf{x})$ et une densité de probabilité *a posteriori* $p(\mathbf{x})$:

$$\alpha = \left[\int \frac{(p(\mathbf{x}))^2}{\tilde{p}(\mathbf{x})} \, d\mathbf{x} \right]^{-1} \tag{5.7}$$

La quantité α peut parfois être vue comme le ratio du volume de la densité *a posteriori* $p(\mathbf{x})$ sur le volume de la densité *a priori* $\tilde{p}(\mathbf{x})$ (par exemple en prenant deux densités uniformes avec le support de la densité *a posteriori* inclus dans celui de la densité *a priori*). Ensuite, les auteurs calquent le modèle de survie 5.7 sur le modèle du filtrage bayésien, en identifiant le rapport $p(\mathbf{x})/\tilde{p}(\mathbf{x})$ comme la vraisemblance, et en approchant la densité *a posteriori* par le nuage de particules $\{\mathbf{x}_t^{(n)}, w_t^{(n)}\}_{n=1}^N$, dont la fonction de proposition est la densité *a priori* $\tilde{p}(\mathbf{x})$. Dans ce cas, les quantités \mathcal{D} et α sont approximativement égales, à un facteur près qui se trouve être le nombre de particules N:

$$\mathcal{D} \simeq \alpha N$$
 (5.8)

Dans un cas idéal, dans lequel tous les poids sont égaux, le diagnostic de survie \mathcal{D} est égal à N, ce qui indique que N particules apportent une information substantielle. Mais en pratique, cette valeur décroît systématiquement (section 1.2.2) et, si sa valeur est trop faible, elle peut être synonyme de divergence du filtre. Ainsi si nous considérons un seul objet et si \mathcal{D}_{min} représente la valeur minimale acceptable du diagnostic de survie, alors pour obtenir l'inégalité $\mathcal{D} \geq \mathcal{D}_{min}$, il faut choisir un nombre de particules tel que $N \geq \mathcal{D}_{min}/\alpha$. Mais lorsque \mathbf{x}_t désigne deux objets, alors le taux de survie défini dans l'équation 5.7 est de α^2 (en prenant deux objets indépendants, et en admettant pour simplifier que chacun d'eux ait le même taux de survie). Cela signifie donc que pour obtenir une même quantité minimale de diagnostic de survie, il faut désormais choisir N tel que $N \geq \mathcal{D}_{min}/\alpha^2$, avec $\alpha \ll 1$. Avec un nombre de M objets, ce résultat se généralise et nécessite de choisir N tel que $N \geq \mathcal{D}_{min}/\alpha^M$, d'où le surcoût exponentiel du nombre de particules par rapport à la dimension de x_t . Pour atténuer ce problème lié à la dimension, les auteurs dans [MacCormick et Isard, 2000] proposent, plutôt que de générer directement des échantillons selon la loi jointe des objets, de partitionner l'espace d'état, et ainsi de manipuler un seul objet à la fois, ce qui permet d'obtenir un taux de survie moins dépendant du nombre d'objets. Des exemples de simulations montrant le bénéfice de cette stratégie d'échantillonnage par rapport à un filtre classique SIR sont exposés dans la thèse [Smith, 2007]. Cette procédure est appelée échantillonnage partitionné et est présentée dans la prochaine section.

5.3 L'échantillonnage partitionné

L'échantillonnage partitionné (PS en anglais, pour *Partitioned Sampling*), introduit dans [MacCormick et Blake, 2000; MacCormick et Isard, 2000], décompose l'espace d'état joint \mathcal{X} en M éléments, formant ainsi une partition (ou produit cartésien) de \mathcal{X} telle que $\mathcal{X} = \mathcal{X}^1 \times \ldots \times \mathcal{X}^M$, avec \mathcal{X}^i l'espace d'état du *i*-ème élément. La construction de la partition est implicite, et consiste seulement à faire correspondre à chaque élément \mathcal{X}^i de la partition un objet \mathbf{x}^i , et pour chacun d'eux, l'algorithme applique la transition (la dynamique) et effectue une procédure de rééchantillonnage pondéré. Cela implique donc qu'il est nécessaire de savoir définir une transition et une fonction de rééchantillonnage pondéré par objet.

Le partitionnement est supposé connu, donc choisi par le concepteur de l'application. Dans un cadre multi-objets, comme nous l'avons déjà signalé, un élément de la partition désigne un seul objet, mais pourrait très bien désigner un groupe d'objets. La stratégie d'échantillonnage partitionné est en fait générale, et ne concerne pas seulement le cadre multi-objets. Elle a par exemple été utilisée dans [Duffner et al., 2009] dans le but d'estimer plusieurs caractéristiques d'un seul objet, en associant un élément de la partition par caractéristique. Dans la suite, nous supposons donc la partition définie, et nous nous concentrons sur la procédure de rééchantillonnage pondéré avant de passer à l'algorithme de l'échantillonnage partitionné.

5.3.1 Rééchantillonnage pondéré

Une procédure de rééchantillonnage pondéré transforme un ensemble de particules $\{\mathbf{x}_{t}^{(n)}, w_{t}^{(n)}\}_{n=1}^{N}$ en un autre tout en laissant la distribution intacte [MacCormick et Blake, 2000; MacCormick, 1999]. Les poids $\{\rho^{(n)}\}_{n=1}^{N}$ sont appelés poids d'importance et sont définis par $\rho^{(n)} = g(\mathbf{x}_{t}^{(n)}) / \sum_{u=1}^{N} g(\mathbf{x}_{t}^{(u)})$. La fonction strictement positive g est appelée fonction de pondération, et a pour but de rééchantillonner les particules sur les « pics » de g. Finalement le nuage de particules $\{\tilde{\mathbf{x}}_{t}^{(n)}, w_{t}^{(n)} / \rho^{(n)}\}_{n=1}^{N}$ est obtenu en sélectionnant les particules avec les probabilités $\{\rho^{(u)}\}_{u=1}^{N}$.

5.3.2 Echantillonnage partitionné

En notant ~ g_i la procédure de rééchantillonnage pondéré du *i*-ème objet, ~ la procédure de rééchantillonnage selon les poids des particules $\{w_t^{(n)}\}_{n=1}^N$, et f_i le processus de transition de l'objet *i*, potentiellement conditionné aux objets déjà générés $\mathbf{x}_t^{1:i-1} \triangleq \{\mathbf{x}_t^k\}_{k=1}^{i-1}$, la procédure de rééchantillonnage pondéré est résumée sur la figure 5.2 et l'algorithme complet est donné dans l'algorithme 5.1.



FIG. 5.2 – Diagramme de la procédure de rééchantillonnage pondéré.

Bien que n'importe quelle fonction g_i laisse la loi *a posteriori* asymptotiquement inchangée, l'objectif de cette étape est d'obtenir une représentation précise de ladite loi. Ainsi, en considérant une factorisation de la vraisemblance telle que $p(\mathbf{y}_t|\mathbf{x}_t) = \prod_{i=1}^{M} p_i(\mathbf{y}_t|\mathbf{x}_t^i)$, alors la vraisemblance $h_i = p_i(\mathbf{y}_t|\mathbf{x}_t^i)$ de l'objet *i* apparaît être un choix **Algorithme 5.1:** Approximation de la loi *a posteriori* $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$ par l'algorithme du filtre particulaire avec échantillonnage partitionné, avec *M* objets.

Entrée : Approximation de la loi *a posteriori* au temps t - 1 :

$$\mathbb{P}(d\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \simeq \sum_{n=1}^{N} w_{t-1}^{(n)} \delta_{\mathbf{x}_{t-1}^{(n)}}(d\mathbf{x}_{t-1})$$

Sortie : Approximation P_N de la loi *a posteriori* au temps $t : \mathbb{P}(d\mathbf{x}_t | \mathbf{y}_{1:t})$ début

- 1. Rééchantillonner $\{\mathbf{x}_{t-1}^{(n)}, 1/N\}_{n=1}^{N}$ à partir du nuage $\{\mathbf{x}_{t-1}^{(n)}, w_{t-1}^{(n)}\}_{n=1}^{N}$ selon l'algorithme du rééchantillonnage multinomial 1.9.
- 2. Echantillonnage partitionné : **pour** i = 1, ..., M 1 **faire**
 - Proposition : pour $n = 1, \dots, N$ faire
 - Générer $\tilde{\mathbf{x}}_{t}^{i,(n)} \sim q(\mathbf{x}_{t}^{i}|\mathbf{x}_{t-1}^{i,(n)}, y_{t})$
 - Rééchantillonnage pondéré : **pour** k = 1, ..., N faire
 - Sélectionner aléatoirement une particule d'indice k selon les probabilités $\{\rho^{(n)}\}_{n=1}^N$ telles que $\forall n = 1, ..., N$:

$$\rho^{(n)} \propto \frac{g_i(\mathbf{x}_t^{i,(n)})}{q(\mathbf{x}_t^i|\mathbf{x}_{t-1}^{i,(n)}, y_t)}$$

• Poser
$$\tilde{\mathbf{x}}_t^{(k)} = \mathbf{x}_t^{(k)}$$

• Poser $\tilde{w}_t^{(k)} = w_t^{(k)} / \rho^{(k)}$

• pour $n = 1, \ldots, N$ faire

• Calculer les poids d'importance normalisés :

$$w_t^{*(n)} = \frac{\tilde{w}_t^{(n)}}{\sum_{m=1}^N \tilde{w}_t^{(m)}}$$

• Copier le nuage des particules $\{\mathbf{x}_t^{(n)}, w_t^{(n)}\}_{n=1}^N = \{\tilde{\mathbf{x}}_t^{(n)}, w_t^{*(n)}\}_{n=1}^N$

- 3. **pour** n = 1, ..., N **faire**
 - Générer pour le dernier objet d'indice $M \mathbf{x}_t^{M,(n)} \sim q(\mathbf{x}_t^M | \mathbf{x}_{t-1}^{M,(n)}, y_t)$
 - Calculer les poids d'importance en utilisant la vraisemblance multiobjets jointe :

$$\tilde{w}_{t}^{(n)} = w_{t}^{(n)} \frac{p(\mathbf{y}_{t}|\mathbf{x}_{t}^{(n)}) \prod_{i=1}^{M} D(\mathbf{x}_{t}^{1:i-1,(n)}, \mathbf{x}_{t-1}^{i,(n)}) \mu_{\mathbf{x}_{t}^{1:i-1,(n)}}(\mathbf{x}_{t}^{i,(n)})}{q(\mathbf{x}_{t}^{M}|\mathbf{x}_{t-1}^{M,(n)}, y_{t})}$$

4. **pour** n = 1, ..., N faire

• Calculer les poids d'importance normalisés :

$$w_t^{(n)} = \frac{\tilde{w}_t^{(n)}}{\sum_{m=1}^N \tilde{w}_t^{(m)}}$$

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

naturel et mène au diagramme proposé sur la figure 5.3, dans lequel nous avons choisi comme fonction de proposition la densité de probabilité *a priori* de la dynamique, soit $f_i = p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i)$. L'indépendance conditionnelle des observations issues du temps *t* sachant les états est en pratique largement répandue, puisqu'elle constitue la manière la plus simple de modéliser une vraisemblance.

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

FIG. 5.3 – Diagramme de la procédure de rééchantillonnage pondéré en utilisant la vraisemblance comme fonction de pondération.

La dépendance entre les états est gérée dans notre modèle par l'introduction de contraintes spatiales floues, que nous avons modélisées dans la section 5.1. Son introduction dans un filtre particulaire multi-objets consiste à intrégrer la densité de probabilité d'interaction définie dans l'équation 5.4 dans le modèle dynamique, selon $f_i = p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i) D(\mathbf{x}_t^{1:i-1}, \mathbf{x}_{t-1}^i) \mu_{\mathbf{x}_t^{1:i-1}}(\mathbf{x}_t^i)$. Ce modèle peut être vu comme le champ de Markov aléatoire utilisé dans [MacCormick et Blake, 1998; Khan et al., 2005; Duffner et al., 2009]. Cependant, dans une perspective plus générale, il est généralement impossible de directement générer des échantillons à partir de $\mu_{\mathbf{x}_t^{1:i-1}}(\mathbf{x}_t^i)$. Ainsi, nous considérons $f_i = p(\mathbf{x}_t^i | \mathbf{x}_{t-1}^i)$ alors que la vraisemblance intègre le terme d'interaction, c'est-à-dire $h_i = p_i(\mathbf{y}_t | \mathbf{x}_t^i) D(\mathbf{x}_t^{1:i-1}, \mathbf{x}_{t-1}^i) \mu_{\mathbf{x}_t^{1:i-1}}(\mathbf{x}_t^i)$. Cette procédure n'affecte par la loi *a posteriori* puisqu'elle peut être vue comme une étape d'échantillonnage par importance [Khan et al., 2005]. Sous ces hypothèses, la procédure d'approximation se simplifie et est donnée dans l'algorithme 5.2.

5.3.3 Discussion

L'échantillonnage partitionné est une méthode vraiment efficace puisqu'en atténuant le problème lié à la dimension, elle réduit de manière significative le coût de calcul. Cependant, comme cela a été discuté dans [Smith et Gatica-Perez, 2004], l'ordre de traitement des objets a un impact direct sur les performances du filtre. Cela est dû aux *M* procédures de rééchantillonnage successives. Ainsi, les objets considérés en premier seront sujets à plus d'effets d'appauvrissement que les autres. D'un autre côté, ceux placés à la fin pourront souffrir d'un manque de diversité avant même d'avoir été considérés, ce qui peut également détériorer les résultats. Pour illustrer le phénomène d'appauvrissement, nous avons choisi d'inclure la figure 5.4 tirée de la thèse de K.Smith [Smith, 2007], qui montre clairement les différences observées entre deux ordres de traitement différents.

La méthode présente une difficulté additionnelle. Si des occultations peuvent survenir, il est généralement plus adéquat de manipuler en premier les objets visibles, en adoptant donc un ordre dynamique de traitement des objets. La solution proposée dans [Mac-Cormick et Blake, 2000] est appelée échantillonnage partitionné à branchements (BPS en anglais, pour *Branched Partitioned Sampling*) et consiste à ajouter au vecteur d'état un vecteur de visibilité, et ainsi à grouper de manière récursive les particules avec une réalisation identique de ce vecteur, ce qui génère implicitement un arbre d'hypothèses. La largeur de cet arbre d'hypothèses aux feuilles peut potentiellement être égale au nombre de permutations possibles du vecteur de visibilité. Ainsi, en considérant un problème de suivi avec M objets, la méthode peut diviser les particules en M! hypothèses. Ce pro**Algorithme 5.2:** Approximation de la loi *a posteriori* $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$ par l'algorithme du filtre particulaire avec échantillonnage partitionné, avec *M* objets, en utilisant comme fonction de pondération attachée au *i*-ème objet sa vraisemblance.

Entrée : Approximation de la loi *a posteriori* au temps t - 1 : $\mathbb{P}(d\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1}) \simeq \sum_{n=1}^{N} w_{t-1}^{(n)} \delta_{\mathbf{x}_{t-1}^{(n)}} (d\mathbf{x}_{t-1})$ Sortie : Approximation P_N de la loi *a posteriori* au temps $t : \mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t})$ début 1. Copier le nuage de particules $\{\mathbf{x}_t^{(n)}, w_t^{(n)}\}_{n=1}^N = \{\mathbf{x}_{t-1}^{(n)}, w_{t-1}^{(n)}\}_{n=1}^N$ 2. pour $i = 1, \dots, M$ faire • Rééchantillonner $\{\mathbf{x}_t^{(n)}, 1/N\}_{n=1}^{N}$ à partir du nuage $\{\mathbf{x}_t^{(n)}, w_t^{(n)}\}_{n=1}^N$ selon l'algorithme du rééchantillonnage multinomial 1.9. • pour $n = 1, \dots, N$ faire • Générer $\mathbf{x}_t^{i,(n)} \sim p(\mathbf{x}_t^{i,(n)})$ • Calculer les poids d'importance : $\tilde{w}_t^{(n)} = p_i(\mathbf{y}_t|\mathbf{x}_t^{i,(n)}) D(\mathbf{x}_t^{1:i-1,(n)}, \mathbf{x}_{t-1}^{i,(n)}) \mu_{\mathbf{x}_t^{1:i-1,(n)}}(\mathbf{x}_t^{i,(n)})$ • pour $n = 1, \dots, N$ faire • Calculer les poids d'importance normalisés : $w_t^{(n)} = \frac{\tilde{w}_t^{(n)}}{\sum_{m=1}^N \tilde{w}_t^{(m)}}$

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



FIG. 5.4 – Figures tirées de [Smith, 2007] montrant le phénomène d'appauvrissement, avec 5 objets A,B,C,D et E, dont l'ordre de traitement est $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$. Les nuages de particules de deux couples d'objets sont projetés dans un plan 2D. Les particules sont représentées par des ronds verts, dont le diamètre est proportionnel à leur poids. (a) Nous pouvons constater un phénomène d'appauvrissement du nuage des particules pour les objets A et B alors que (b) le nuage de particules pour les objets D et E est varié, grâce à leur position tardive dans l'ordre de traitement des objets.

cédé contient deux inconvénients majeurs. Premièrement, l'algorithme peut propager des particules dans des sous-arbres de l'arbre d'hypothèses dont le vecteur de visibilité est erroné, sans jamais pouvoir se confronter aux particules des autres sous-arbres d'hypothèses. Chaque sous-arbre opère donc en aveugle, ce qui peut mener à une mauvaise estimation de la loi jointe *a posteriori*. De plus, en divisant le nombre N de particules en M! hypothèses, cela signifie que le nombre de particules alloué par chemin est potentiellement de l'ordre de N/M! éléments. Ainsi, les particules n'essaient plus de survivre parmi un ensemble de N éléments mais parmi des ensembles potentiellement non pertinents de N/M! éléments, qui regroupent moins de diversité et diminuent donc les chances d'obtenir des échantillons de forte vraisemblance.

L'échantillonnage partitionné dynamique (DPS en anglais, pour *Dynamic Partitioned Sampling*), proposé dans [Smith et Gatica-Perez, 2004], utilise un mélange de lois pour représenter la distribution *a posteriori*. Chaque composante représente un ordre spécifique de traitement des objets. Dans leurs expériences, les auteurs utilisent M ensembles de permutations, définis de manière déterministe (en posant un ordre puis en opérant un changement cyclique de manière à n'affecter qu'une seule fois un objet à une position), chacun possédant N/M particules. Cette stratégie améliore les résultats de l'échantillonnage partitionné puisqu'elle diminue le phénomème d'appauvrissement, ce qui se constate particulièrement lorsque des occultations surviennent. Cependant, utiliser un petit nombre de sous-ensembles de permutations possibles fixés peut ne pas être robuste. De plus, diviser les particules en plusieurs ensembles possède les mêmes inconvénients que l'échantillonnage partitionné à branchements, puisque les particules évoluent dans des sous-ensembles différents.

Afin de surmonter ces problèmes, nous proposons une nouvelle stratégie d'échantillonnage, appelée échantillonnage partitionné à rang (RPS en anglais, pour *Ranked Partitioned Sampling*), qui estime de manière jointe l'état des objets ainsi que leur ordre de traitement. Les objets avec les confiances les plus fortes seront considérés dans les premières étapes du scénario, car ce sont ceux qui sont les plus aptes à surmonter un phénomène d'appauvrissement des particules. Chaque scénario est ainsi confronté à chacun des autres, ce qui permet d'implicitement élaguer ceux qui ne seront pas pertinents. Le choix adaptatif de l'ordre de traitement a pour but de limiter le phénomène d'appauvrissement.

5.4 Echantillonnage partitionné à rang

Soit $\mathbf{o}_t = (\mathbf{o}_t^1, \dots, \mathbf{o}_t^M)$ un ordre de traitement, c'est-à-dire une permutation sur M objets. La variable \mathbf{o}_t^i indique la position de l'objet i dans cette séquence. Nous appelons un scénario la permutation réciproque \mathbf{o}_t^{-1} considérée à un instant particulier t, que nous notons $\mathbf{s}_t = (\mathbf{s}_t^1, \dots, \mathbf{s}_t^M)$. Ainsi, la k-ième composante d'un scénario est définie telle que $\mathbf{s}_t^k = i$ si et seulement si $\mathbf{o}_t^i = k$ (ou de manière équivalente, $\mathbf{s}_t^k \triangleq \sum_{i=1}^M i \, \delta_{\mathbf{o}_t^i}^k$, avec δ_a^b la fonction de Kronecker qui vaut 1 si a = b, 0 sinon).

Par exemple, considérons quatre objets et un ordre de traitement tel que o = (2, 3, 4, 1). Cela signifie que l'objet d'identifiant 1 sera considéré en 2-ème position, l'objet 2 en 3-ème, etc. Le scénario correspond alors à s = (4, 1, 2, 3) et indique de manière ordonnée les identifiants des objets dans le processus séquentiel d'approximation (objet 4 en première position, etc.).

Nous considérons tout d'abord les probabilités fixées de transition des positions :

$$\mathbb{P}(\mathbf{s}_{t}^{h} = i | \mathbf{s}_{t-1}^{k} = i) \triangleq \mathbb{P}(\mathbf{o}_{t}^{i} = h | \mathbf{o}_{t-1}^{i} = k) \triangleq \alpha_{k,h} \quad \forall i \in \{1, \dots, M\}$$
(5.9)

La modélisation de l'ordre de traitement \mathbf{o}_t est nécessaire car nous souhaitons modéliser la transition $\mathbb{P}(\mathbf{o}_t^i = h | \mathbf{o}_{t-1}^i = k)$, qui indique la probabilité de l'objet *i* de passer d'une position *k* à une position *h*. A l'inverse, l'information de transition du scénario a moins de sens, puisque $\mathbb{P}(\mathbf{s}_t^k = i | \mathbf{s}_{t-1}^k = j)$ représente la probabilité que la position *k* passe de l'objet *j* à l'objet *i*, sur laquelle nous ne pouvons apporter d'*a priori* fort. En revanche, le vecteur \mathbf{s}_t nous est utile pour désigner l'indice de l'objet à considérer à un instant *k* du processus séquentiel d'estimation.

L'utilisation d'une matrice de transition fixée permet de poser de manière intuitive les probabilités de transition du processus discret. Pour la modélisation de la matrice de transition de l'ordre de traitement, nous privilégierons les états statiques, ainsi la matrice déterminée par les coefficients $\alpha_{k,h}$ devrait avoir des valeurs maximales sur sa diagonale, sans être systématiquement symétrique. Nous appelons Q_{α} cette matrice, telle que $Q_{\alpha} =$ $(\alpha_{k,h})_{(k,h)\in\{1,...,M\}^2}$. En considérant en premier les objets placés dans les premières étapes au temps t - 1, la distribution jointe de transition de \mathbf{o}_t s'écrit :

$$\mathbb{P}(\mathbf{o}_t|\mathbf{o}_{t-1}) = \mathbb{P}\left(\mathbf{o}_t^{s^1}|\mathbf{o}_{t-1}^{s^1}\right) \prod_{k=2}^M \mathbb{P}\left(\mathbf{o}_t^{s^k}|\mathbf{o}_{t-1}^{s^k} \triangleq k, \mathbf{o}_t^{s^1}, \dots, \mathbf{o}_t^{s^{k-1}}\right)$$
(5.10)

avec $\mathbf{s}^k \triangleq \mathbf{s}_{t-1}^k$, l'indice du temps étant omis pour simplifier les notations.

En reprenant l'exemple avec $\mathbf{o} = (2, 3, 4, 1)$ et en considérant que cette estimation a eu lieu au temps t - 1, l'équation 5.10 nous donne :

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

La décomposition de la loi jointe $\mathbb{P}(\mathbf{o}_t | \mathbf{o}_{t-1})$ est donc déterminée par l'estimation à t - 1 du scénario, $\mathbf{s}_{t-1} = (4, 1, 2, 3)$.

La dernière distribution conditionnelle dans l'équation 5.10 dépend des probabilités de transition des positions définies dans l'équation 5.9 ainsi que des positions inaccessibles des objets déjà considérés :

$$\mathbb{P}\left(\mathbf{o}_{t}^{s^{k}} = h | \mathbf{o}_{t-1}^{s^{k}} \triangleq k, \mathbf{o}_{t}^{s^{1}}, \dots, \mathbf{o}_{t}^{s^{k-1}}\right) = \left[1 - \sum_{j=1}^{k-1} \delta_{\mathbf{o}_{t}^{s^{j}}}^{h}\right] \left[\alpha_{k,h} + \frac{1}{M-k+1} \sum_{j=1}^{k-1} \alpha_{k,\mathbf{o}_{t}^{s^{j}}}\right] (5.11)$$

avec $\mathbf{s}^k \triangleq \mathbf{s}_{t-1}^k$. Le premier terme dans le produit garantit que la probabilité est nulle si la position h a déjà été assignée, et vaut 1 dans le cas contraire. Le second terme contient les probabilités de transition *a priori* d'une position h à k, telles qu'elles ont été définies dans l'équation 5.9 ainsi qu'un terme qui utilise les probabilités de transition des positions assignées pour équilibrer la distribution (ce sont les positions dont les objets ont déjà été traités, $\mathbf{o}_t^{s^1}, \ldots, \mathbf{o}_t^{s^{k-1}}$), et redistribue ces probabilités d'une manière uniforme à toutes les positions qui n'ont pas été encore assignées (toutes celles dont le premier terme vaut 1). Comme k - 1 positions ont déjà été affectées, il en reste M - (k - 1) disponibles, ce qui explique le dénominateur du facteur de la dernière somme.

Pour faciliter la compréhension de la formule 5.11, reprenons l'exemple avec $o_{t-1} = (2, 3, 4, 1)$ et $s_{t-1} = (4, 1, 2, 3)$. Les équations s'écrivent alors :

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

Vérifions que la distribution 5.11 est correcte en testant si elle somme à 1 :

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

Nous utilisons la décomposition suivante de la densité de transition jointe $p(\mathbf{x}_t, \mathbf{o}_t | \mathbf{x}_{t-1}, \mathbf{o}_{t-1})$:

$$p(\mathbf{x}_t, \mathbf{o}_t | \mathbf{x}_{t-1}, \mathbf{o}_{t-1}) = p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{o}_t) \mathbb{P}(\mathbf{o}_t | \mathbf{o}_{t-1})$$
(5.12)

avec $p(\mathbf{o}_t | \mathbf{o}_{t-1})$ la loi de transition définie dans l'équation 5.10. Conditionnée par l'ordre de traitement défini par \mathbf{o}_t , la densité de transition du vecteur d'état \mathbf{x}_t est décomposée en considérant en premier les objets placés dans les premières étapes :

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{o}_t) \triangleq \prod_{k=1}^M p\left(\mathbf{x}_t^{s_t^k} | \mathbf{x}_{t-1}^{s_t^k}, \mathbf{x}_t^{s_t^1}, \dots, \mathbf{x}_t^{s_t^{k-1}}\right)$$
(5.13)

Ainsi, l'ordre de la chaîne de Markov définie par les composantes de \mathbf{x}_t , c'est-à-dire les composantes des objets $\{\mathbf{x}_t^i\}_{i=1}^M$, est déterminé par le processus discret d'ordre de traitement des objets \mathbf{o}_t , ou, de manière équivalente, par le processus discret représentant un scénario \mathbf{s}_t .

Reprenons l'exemple du scénario $s_{t-1} = (4, 1, 2, 3)$, et imaginons que sa transition à t soit égale à $s_t = (4, 2, 1, 3)$, alors s_t définit la décomposition de la densité de probabilité jointe de la dynamique de l'état 5.13 :

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

Cependant, en suivant les choix faits dans la section 5.3, nous posons le processus dynamique tel que

$$f_{\mathbf{s}_{t}^{k}} = p\left(\mathbf{x}_{t}^{s_{t}^{k}} | \mathbf{x}_{t-1}^{s_{t}^{k}}, \mathbf{x}_{t}^{s_{t}^{1}}, \dots, \mathbf{x}_{t}^{s_{t}^{k-1}}\right) = p\left(\mathbf{x}_{t}^{\mathbf{s}_{t}^{k}} | \mathbf{x}_{t-1}^{\mathbf{s}_{t}^{k}}\right)$$
(5.14)

ce qui signifie qu'un objet n'est conditionné que par lui-même, à l'instant précédent. Ainsi, la vraisemblance s'écrit :

$$p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{o}_t) \triangleq \prod_{k=1}^M p_{\mathbf{s}_t^k} \left(\mathbf{y}_t | \mathbf{x}_t^{\mathbf{s}_t^k} \right)$$
(5.15)

Pour résumer, pour chaque particule, à un instant t, l'algorithme génère en premier lieu l'ordre de traitement des objets, et donc le scénario. Ensuite, à chaque étape k du processus de partitionnement, l'algorithme rééchantillonne le nuage de particules, avant de proposer un nouvel état de l'objet désigné par le scénario à la position k, puis calcule sa vraisemblance. L'approximation de la loi de filtrage jointe de $(\mathbf{x}_t, \mathbf{o}_t)$ est obtenue une fois les M opérations effectuées. En fixant la vraisemblance finale, qui prend en compte les interactions entre objets, soit $h_{\mathbf{s}_t^k} = p_{\mathbf{s}_t^k}(\mathbf{y}_t | \mathbf{x}_t^{\mathbf{s}_t^k}) D(\mathbf{x}_t^{\mathbf{s}_t^{1:\mathbf{s}_t^{k-1}}}, \mathbf{x}_{t-1}^{\mathbf{s}_t^k}) \mu_{\mathbf{x}_t^{\mathbf{s}_t^{1:\mathbf{s}_t^{k-1}}}}(\mathbf{x}_t^{\mathbf{s}_t^k})$, nous obtenons le diagramme de la figure 5.5 ainsi que la procédure d'approximation décrite dans l'algorithme 5.3.



FIG. 5.5 – Diagramme de la procédure d'échantillonnage partitionné à rang en utilisant la vraisemblance comme fonction de pondération.

Il existe plusieurs manières de gérer les objets cachés. Un choix courant est d'estimer le vecteur de visibilité et de proposer un scénario selon celui-ci. Cependant, pour l'échantillonnage partitionné à rang et comme nous le verrons dans la section 5.5.2, nous considérons implicitement qu'un objet à une position k est toujours moins visible qu'un autre à une position k - h, avec 0 < h < k. Intégrer de manière explicite le vecteur de visibilité dans l'échantillonnage partitionné à rang est toutefois possible, et requiert d'adapter la distribution de génération des scénarios 5.11, en la conditionnant par le vecteur de visibilité. Toutefois, l'avantage d'une modélisation implicite, est qu'elle ne rajoute pas de dimension supplémentaire à l'espace d'état.
Algorithme 5.3: Approximation de la loi *a posteriori* $\mathbb{P}(d\mathbf{x}_t | \mathbf{y}_{1:t})$ par l'algorithme du filtre particulaire avec échantillonnage partitionné à rang, avec M objets, en utilisant comme fonction de pondération attachée au *i*-ème objet sa vraisemblance.

Entrée : Approximation de la loi *a posteriori* au temps t - 1 :

$$\mathbb{P}(d\mathbf{x}_{t-1}, \mathbf{o}_{t-1} | \mathbf{y}_{1:t-1}) \simeq \sum_{n=1}^{N} w_{t-1}^{(n)} \, \delta_{\mathbf{o}_{t-1}^{(n)}}^{\mathbf{o}_{t-1}} \, \delta_{\mathbf{x}_{t-1}^{(n)}}^{(n)} (d\mathbf{x}_{t-1})$$

Sortie : Approximation P_N de la loi *a posteriori* au temps $t : \mathbb{P}(d\mathbf{x}_t, \mathbf{o}_t | \mathbf{y}_{1:t})$ début

- 1. Copier le nuage de particules $\{\mathbf{x}_{t}^{(n)}, w_{t}^{(n)}\}_{n=1}^{N} = \{\mathbf{x}_{t-1}^{(n)}, w_{t-1}^{(n)}\}_{n=1}^{N}$
- 2. Générer l'ordre de traitement des objets et le scénario : **pour** n = 1, ..., N faire

 - Générer o_t⁽ⁿ⁾ ~ P(o_t|o_{t-1}⁽ⁿ⁾) selon l'équation 5.10
 Calculer le scénario s_t⁽ⁿ⁾ tel que ∀i = 1,..., M, s_t^{i,(n)} = ∑_{k=1}^M k δ_{o_t^{k,(n)}}ⁱ

3. pour $k = 1, \ldots, M$ faire

- Rééchantillonner $\{\mathbf{x}_t^{(n)}, 1/N\}_{n=1}^N$ à partir du nuage $\{\mathbf{x}_t^{(n)}, w_t^{(n)}\}_{n=1}^N$ selon l'algorithme du rééchantillonnage multinomial 1.9.
- pour $n = 1, \ldots, N$ faire

• Générer
$$\mathbf{x}_{t}^{\mathbf{s}_{t}^{k,(n)},(n)} \sim p(\mathbf{x}_{t}^{\mathbf{s}_{t}^{k,(n)}} | \mathbf{x}_{t-1}^{\mathbf{s}_{t}^{k,(n)},(n)})$$

• Calculer les poids d'importance :

$$\tilde{w}_{t}^{(n)} = p_{i} \left(\mathbf{y}_{t} | \mathbf{x}_{t}^{\mathbf{s}_{t}^{k,(n)},(n)} \right) D \left(\mathbf{x}_{t}^{\mathbf{s}_{t}^{1,(n)}:\mathbf{s}_{t}^{k-1,(n)},(n)}, \mathbf{x}_{t-1}^{\mathbf{s}_{t}^{k,(n)},(n)} \right) \mu_{\mathbf{x}_{t}^{\mathbf{s}_{t}^{1,(n)}:\mathbf{s}_{t}^{k-1,(n)},(n)} \left(\mathbf{x}_{t}^{\mathbf{s}_{t}^{k,(n)},(n)} \right)$$

• pour $n = 1, \ldots, N$ faire

• Calculer les poids d'importance normalisés :

$$w_t^{(n)} = \frac{\tilde{w}_t^{(n)}}{\sum_{m=1}^N \tilde{w}_t^{(m)}}$$

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

5.5 Expériences

5.5.1 Description des séquences

Pour valider expérimentalement les propositions effectuées dans ce chapitre, nous proposons trois expériences, visant à montrer :

- l'intérêt des contraintes spatiales floues en les comparant à une approche sans contrainte (sections 5.5.3 et 5.5.4);
- l'intérêt de l'échantillonnage partitionné à rang (RPS), associé à la notion de visibilité implicite induite par le scénario, en le comparant à l'approche classique de l'échantillonnage partitionné à branchements (BPS) (section 5.5.2);
- la stabilité du RPS lorsque tous les scénarios possibles semblent équivalents (section 5.5.3);
- l'intérêt du RPS en le comparant à l'approche classique de l'échantillonnage partitionné (PS) (section 5.5.4).

5.5.1.1 Suivi de piétons

L'objectif de cette première expérience est de suivre des piétons se déplaçant dans un environnement intérieur (figure 5.6). De cette séquence publique disponible sur [Fleuret et al., 2008], nous avons extrait 280 images dans lesquelles trois piétons marchent et s'occultent. Nous considérons que le déplacement et la position de chacun des piétons sont indépendants de ceux des autres, ainsi aucune contrainte spatiale n'est modélisée pour cette séquence. Le but est donc de tester le comportement de l'algorithme de l'échantillonnage partitionné à rang, en particulier par la notion de visibilité implicite dont il dépend, et de comparer ses résultats à une méthode standard, celle de l'échantillonnage partitionné à branchements (BPS), qui est une extension du PS dans le cas où des objets peuvent s'occulter.



FIG. 5.6 – Image de la séquence de piétons.

5.5.1.2 Suivi de fourmis

Nous proposons ensuite de suivre des fourmis dans une séquence d'images dont la vue est en plongée (figure 5.7). Cette séquence a été proposée dans [Khan et al., 2005], en utilisant un filtre particulaire MCMC, et a également été utilisée avec un filtre particulaire multi-objets énergétique dans [El Abed et al., 2007; El Abed, 2008]. La séquence contient 750 images représentant de 4 à 8 fourmis. Les entrées et sorties des fourmis se font par un trou au centre de la scène, et sont gérées manuellement dans nos algorithmes de suivi (c'est-à-dire en définissant explicitement les processus de naissance et de mort de chaque fourmi). La position de la caméra permet de considérer que les fourmis ne

peuvent pas se chevaucher, et nous exploitons ainsi cette hypothèse en modélisant une contrainte spatiale d'exclusion.



FIG. 5.7 – Image de la séquence de fourmis.

5.5.1.3 Suivi de main

Nous proposons une dernière séquence de 800 images de suivi de la main que nous avons créée pour les besoins de nos tests. Les doigts peuvent n'être que partiellement visibles, ce qui constitue la principale difficulté de cette expérience (figure 5.8). Nous proposons de contraindre le positionnement des quatre doigts à estimer (l'index, le majeur, l'annulaire et l'auriculaire) en utilisant les contraintes spatiales floues de distance, d'angle, d'alignement et d'exclusion. La forme globale de l'estimation ainsi contrainte permet d'estimer correctement la position centrale et l'angle de tous les doigts, même s'ils sont partiellement cachés.



FIG. 5.8 – Image de la séquence de la main.

5.5.2 Suivi de piétons

Soit $\mathbf{x}_t^i = (\mathbf{x}_t^i, \mathbf{y}_t^i)^T$ l'état inconnu de l'objet *i* (c'est-à-dire une personne), avec $(\mathbf{x}_t^i, \mathbf{y}_t^i)^T$ son centre 2*D*. Les dimensions des rectangles englobant les individus sont fixées. La dynamique est une marche aléatoire, qui s'écrit $\mathbf{x}_t^i = f_t(\mathbf{x}_{t-1}^i, \mathbf{v}_t^i) = \mathbf{x}_{t-1}^i + \mathbf{v}_t^i$ avec \mathbf{v}_t^i un bruit blanc gaussien, de moyenne nulle et de matrice de covariance diagonale $\sigma_x^2 = \sigma_y^2 = 4^2$. La matrice de transition Q_α est fixée à :

$$Q_{\alpha} = \begin{bmatrix} 0.6 & 0.3 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}$$
(5.16)

afin de privilégier le fait qu'un objet reste à la même position dans l'ordre de traitement. Comme nous pouvons le constater, la matrice Q_{α} n'est pas symétrique. En effet, pour les positions intermédiaires, disons 1 < k < M), nous souhaitons poser une probabilité identique de passer à la place k - i, ou de passer à la place k + i, avec $i < \min(k, \leq M - k + 1)$. Cela, afin qu'un objet ait une équiprobabilité de gagner, ou de perdre, une place (ou plusieurs) dans le scénario. Cela n'est pas vrai pour les positions excentrées, qui possèdent moins de possibilités de changement de position, c'est pourquoi nous augmentons leurs probabilités de perdre, ou de gagner (selon si la position considérée est la première, ou la dernière), une place (ou plusieurs), dans le scènario, ce qui explique les valeurs de la matrice Q_{α} .

Pour le BPS, nous modélisons le vecteur de visibilité, où la probabilité de transition de l'état visible à celui de caché est fixée à 0.2, et la probabilité de transition de l'état caché à celui de visible à 0.5. Pour le RPS, nous considérons implicitement la visibilité d'un objet par sa position dans l'ordre de traitement, ainsi aucune contrainte spatiale n'est nécessaire dans cette expérience. Pour les deux méthodes, les objets visibles sont considérés en premier, bien que dans le RPS le vecteur de visibilité ne soit pas modélisé puisque c'est le vecteur de scénario qui détermine la visibilité partielle d'un objet. La vraisemblance d'un objet est une distance entre histogrammes de couleurs, comme cela a déjà été étudié dans la section 4.1.3. Cependant, seules les parties visibles de l'objet sont considérées dans le but de ne pas pénaliser les objets cachés ou partiellement cachés (sont considérées cachées les parties de l'objet qui correspondent à des régions communes à d'autres objets estimés avant dans l'ordre de traitement). En notant a_t^i l'ensemble des pixels couverts par le rectangle de l'objet *i* à un instant *t*, définie par son état \mathbf{x}_{t}^{i} , $h_{i}^{\bullet} = \{h_{i}^{\bullet}(u)\}_{u=1}^{U}$ le modèle d'histogramme appris de l'objet i (en segmentant l'objet manuellement sur la première image), et $h_{BG}^{\bullet} = \{h_{BG}^{\bullet}(u)\}_{u=1}^{U}$ le modèle d'histogramme appris de l'image de fond (en définissant manuellement une région d'intérêt dans la première image), et ul'indice d'une case d'un histogramme de longueur U, la vraisemblance s'écrit :

$$p(\mathbf{y}_{t}|\mathbf{x}_{t}^{\mathbf{s}_{t}^{k}}, \mathbf{x}_{t}^{\mathbf{s}_{t}^{1}:\mathbf{s}_{t}^{k-1}}) = p(\mathbf{y}_{t}|\breve{a}_{t}^{\mathbf{s}_{t}^{k}} = \left\{ \mathring{a}_{t}^{\mathbf{s}_{t}^{k}} \setminus \bigcup_{h=1}^{k-1} \mathring{a}_{t}^{\mathbf{s}_{t}^{h}} \right\})$$

$$\propto \exp\left(-\lambda \left(d^{2} \left[h_{\mathbf{s}_{t}^{\mathbf{s}_{t}^{k}}}, h(\breve{a}_{t}^{\mathbf{s}_{t}^{k}})\right] - d^{2} \left[h_{BG}^{\mathbf{s}_{t}}, h(\breve{a}_{t}^{\mathbf{s}_{t}^{k}})\right] \right) \right)$$
(5.17)

avec la constante multiplicative $\lambda = 30$ fixée empiriquement, d^2 la distance de Bhattacharyya définie dans l'équation 4.12 et $a_t^{\mathbf{s}_t^k}$ les pixels visibles de l'objet d'indice \mathbf{s}_t^k , conditionnellement aux objets d'indices $\{\mathbf{s}_t^h\}_{h=1}^{k-1}$. Un exemple de la vraisemblance obtenue, dans le cas d'une occulation, est donnée sur la figure 5.9. Lorsque l'on considère qu'un objet est toujours visible (figure 5.9(b)), les valeurs obtenues à la vraie position de l'objet sont très atténuées, en raison de la présence d'un autre objet occultant celui-ci. De ce fait, les poids obtenus favoriseront les localisations excentrées, qui contiennent une partie du vrai objet. Dans la figure 5.9(c), dans laquelle nous considérons que l'objet bleu est traité en premier, donc plus visible que l'objet détouré en vert, ce dernier ne prend pas en compte les pixels appartenant à l'aire bleue, ce qui se traduit par une meilleure localisation des fortes densités de la vraisemblance.

Les résultats avec N = 500 particules sont illustrés sur la figure 5.10. Les rectangles en rouge symbolisent l'estimation des objets (c'est-à-dire l'espérance de Monte-Carlo). Comme mentionné dans la section 5.3.3, dans le BPS, les particules peuvent être divisées en M! ensembles, qui peuvent correspondre à des scénarios dans lesquels les hypothèses de visibilité sont fausses. De plus, le vecteur de visibilité n'est pas bien adapté dans le cas où le nombre d'objets est supérieur à deux, puisqu'il ne résout plus le problème d'association aux données. Ces deux points expliquent la différence entre les résultats obtenus par le BPS et ceux du RPS (voir par exemple les deuxième et dernière images).

Les figures 5.11(b) et 5.11(c) montrent les erreurs quadratiques moyennes (RMSE) ob-



FIG. 5.9 – Tests sur la log-vraisemblance dans le cas d'une occultation partielle utilisée dans la séquence de piétons. (a) Image test, dans laquelle nous souhaitons trouver l'individu désigné par un rectangle vert alors qu'il est occulté par l'individu en bleu, puis les vraisemblances obtenues pour l'individu dont le rectangle est vert (b) sans utiliser de masque, et (c) en utilisant un masque (dans ce cas, les pixels appartenant à la région bleue ne sont pas pris en compte). Une couleur rouge indique une forte vraisemblance, une couleur noire désigne la valeur maximale atteinte par la vraisemblance.



FIG. 5.10 – Résultats du suivi de piétons aux temps 5, 24, 51, 99, 120, 205, 212 et 259. (a) BPS, (b) RPS. Le RPS gère mieux les occultations, comme par exemple aux temps 24 et 212.

tenues pour chacun des piétons et pour les méthodes RPS et BPS (voir la section 1.2.4 pour le calcul du RMSE). Nous pouvons y constater la supériorité globale du RPS sur le BPS. La figure 5.11(d) présente les probabilités *a posteriori* obtenues par le RPS pour une personne d'être considérée en premier dans l'ordre de traitement des objets induit par o_t . Une probabilité faible indique que la personne peut être partiellement cachée. Nous pouvons remarquer que les probabilités estimées par exemple aux temps 24, 99, 205 et 212 sont cohérentes avec la séquence (figure 5.10). C'est ce que l'on peut particulièrement observer aux temps 205 à 212 (sixième et septième images de la figure 5.10(b)), durant lesquels l'individu d'indice 3 passe devant celui d'indice 2, lui-même occulant le dernier piéton. Jusqu'à la fin de la séquence (dernière image de la figure 5.10(b)), l'individu d'indice 3 s'ecarte tandis que celui d'indice 2 reste devant le dernier, ce qui reste cohérent avec le changement de probabilités obtenues dans la figure 5.11(d), qui place l'individu d'indice 3 au temps 205 en premier, avant de privilégier celui d'indice 2 à partir du temps 212.



FIG. 5.11 – (a) Indices des piétons présents dans la scène, (b) l'erreur RMSE du BPS, (c) l'erreur RMSE du RPS (l'approche proposée) et (d) les probabilités *a posteriori* obtenues par le RPS pour une personne d'être considérée en premier dans l'ordre de traitement des objets.

5.5.3 Suivi de fourmis

L'état de l'objet *i*, $\mathbf{x}_t^i = (\mathbf{x}_t^i, \mathbf{y}_t^i, \theta_t^i)^T$, contient sa position $2D (\mathbf{x}_t^i, \mathbf{y}_t^i)^T$ et son orientation θ_t^i . Les dynamiques de position et d'orientation sont des marches aléatoires. Les dimensions des rectangles des fourmis sont fixées (12 pixels de largeur et 40 de longueur pour toutes les fourmis). Le paramètre de bruit de position est de moyenne nulle et sa matrice de covariance est diagonale avec $\sigma_x^2 = \sigma_y^2 = 11^2$ (ces valeurs sont légèrement plus élevées que celles utilisées dans [Khan et al., 2005], sans que son influence sur les résultats soit réellement significative). Le paramètre de bruit d'orientation est de moyenne nulle et de

variance $\sigma_{\theta}^2 = 0.3^2$. La matrice de transition Q_{α} est fixée à :

	0.5	0.2	0.1	0.07	0.05	0.03	0.03	0.02	
$Q_{lpha} =$	0.15	0.43	0.15	0.1	0.07	0.05	0.03	0.02	(5.18
	0.09	0.14	0.39	0.14	0.09	0.07	0.05	0.03	
	0.06	0.085	0.14	0.39	0.14	0.085	0.06	0.04	
	0.04	0.06	0.085	0.14	0.39	0.14	0.085	0.06	
	0.03	0.05	0.07	0.09	0.14	0.39	0.14	0.09	
	0.02	0.03	0.05	0.07	0.1	0.15	0.43	0.15	
	0.02	0.03	0.03	0.05	0.07	0.1	0.2	0.5	

Les probabilités $(\alpha_{k,h})_{(k,h)\in\{1,\ldots,M\}^2}$ ont été fixées à la main, de manière à donner une souplesse dans la transition des positions, dans le but de favoriser la diversification d'hypothèses, c'est pourquoi les coefficients diagonaux ne sont pas égaux (il est en effet logique de considérer que des fourmis traitées en milieu de traitement à t - 1 ont une probabilité importante de changer radicalement de position, ce qui est évidemment moins le cas avec celles traitées en début ou fin d'ordre de traitement). La précision de ces probabilités est due à la normalisation. Comme celle définie dans l'équation 5.16, la matrice Q_{α} n'est pas symétrique, pour les mêmes raisons que celles invoquées dans la section 5.5.2.

Nous avons utilisé pour cette expérience une contrainte spatiale d'exclusion : en utilisant un semi-trapèze flou, deux fourmis ne peuvent se superposer à plus de 10% de leurs aires, et à partir de 5% le degré de satisfaction de la contrainte commence à décroître (figure 5.12(a)). La densité de probabilité de la contrainte spatiale est donc définie de la manière suivante :

$$\mu_{\tilde{\mathbf{x}}_{t}}(\mathbf{x}_{t}^{m}) = \nu_{\tilde{\mathbf{x}}_{t}}(\mathbf{x}_{t}^{m})^{\gamma}$$

$$\nu_{\tilde{\mathbf{x}}_{t}}(\mathbf{x}_{t}^{m}) = \frac{E}{k=1} \nu_{\tilde{\mathbf{x}}_{t}}^{k}(\mathbf{x}_{t}^{m})$$

$$= {}^{B}\nu_{\tilde{\mathbf{x}}_{t}}^{exclusion}(\mathbf{x}_{t}^{m})$$

$$B^{exclusion}(\mathbf{x}_{t}^{m}) = \frac{E}{\psi_{t=1}} \nu_{\tilde{\mathbf{x}}_{t}}^{exclusion}(\mathbf{x}_{t}^{m})$$

$$= \min_{l=1,\dots,L} {}^{B}\nu_{\tilde{\mathbf{x}}_{l}}^{exclusion}(\mathbf{x}_{t}^{m})$$

La contrainte spatiale d'exclusion $\mu_{\tilde{\mathbf{x}}_t}^{exclusion}(\mathbf{x}_t^m)$, par rapport au point central d'un objet i ($\mathbf{x}_t^i, \mathbf{y}_t^i$), en considérant trois fourmis déjà traitées, est calculée sur toute l'image et représentée sur la figure 5.12(b).

La vraisemblance utilisée utilise une simple soustraction de fond. Nous notons a_t^i l'ensemble des pixels couverts par le rectangle de la fourmi *i* à un instant *t*, $I_{BG} : \Omega \to [0, 255]$ l'image en niveaux de gris représentant le fond de l'image (image moyenne calculée sur plusieurs images consécutives ne présentant pas de fourmis), $I_t : \Omega \to [0, 255]$ l'image en niveaux de gris au temps *t*, alors la vraisemblance de la fourmi *i* s'écrit :

$$p(\mathbf{y}_t | \mathbf{x}_t^i) \propto \exp\left(\frac{\lambda}{P_{\mathring{a}_t^i}} \operatorname{card}\left(\mathring{a}_t^i \cap P_{\backslash BG}\right)\right)$$
(5.19)

avec $\lambda = 10$ la constante multiplicative fixée empiriquement, $P_{\hat{a}_t^i}$ le nombre de pixels couverts par l'ensemble \hat{a}_t^i , $P_{\backslash BG} = \{p; |I_{BG}(p) - I_t(p)| > T, p \in \Omega\}$ l'ensemble des pixels



FIG. 5.12 – (a) Fonction décroissante représentant le degré de satisfaction de la contrainte spatiale d'exclusion ${}^{B}\mu^{exclusion}$ en fonction de l'aire de recouvrement entre \mathbf{x}_{t}^{m} et $\tilde{\mathbf{x}}_{t}^{l}$, et (b) Contrainte spatiale obtenue sur toute l'image, en considérant trois fourmis déjà traitées en rouge, et une fourmi dont l'orientation est superposée à l'image de contrainte, en jaune. Un niveau de gris bas indique que la contrainte est satisfaite.

qui diffèrent de plus de T en valeur absolue de l'image de fond apprise (nous avons fixé le seuil T = 20 de manière empirique). La vraisemblance obtenue est illustrée sur la figure 5.13, avec quatre fourmis. La vraisemblance étant la même pour chacune d'elle, une contrainte d'exclusion est clairement nécessaire pour éviter qu'elles se confondent.

La figure 5.14 montre les résultats du RPS sans et avec contrainte spatiale, et du PS avec contrainte spatiale. Les positions estimées des fourmis sont représentées en rouge. N = 500 particules ont été utilisées dans cette séquence. Le bénéfice d'utiliser une contrainte spatiale simple est très clair ici (plusieurs fourmis ne sont pas suivies sans contrainte spatiale). Le PS et le RPS donnent des résultats comparables puisque tous les ordres de traitement donnent des résultats quasi identiques. Donc le RPS accomplit la tâche aussi bien que le PS lorsque l'ordre importe peu et que le nombre d'objets est assez petit pour ne pas souffrir d'un phénomène d'appauvrissement.

5.5.4 Suivi de main

Nous considérons finalement un problème de suivi d'un objet articulé, la main droite (spécifier la main n'est pas sans intérêt, puisque nous définirons les positions relatives des doigts). L'état de l'objet i, $\mathbf{x}_t^i = (\mathbf{x}_t^i, \mathbf{y}_t^i, \theta_t^i)^T$, contient sa position 2D centrale $(\mathbf{x}_t^i, \mathbf{y}_t^i)^T$ et son orientation θ_t^i . Chaque forme de doigt est fixée et représentée par une courbe B-spline définie par un vecteur de 6 points de contrôle 2D, localisés sur la base des doigts, au milieu et au bout des doigts. Nous considérons que les formes des doigts sont rigides, ainsi les points de contrôle sont fixés. Les dynamiques de position et d'orientation sont des marches aléatoires. Le paramètre de bruit de position est de moyenne nulle et sa matrice de covariance est diagonale avec $\sigma_x^2 = \sigma_y^2 = 8^2$. Le paramètre de bruit d'orientation est de moyenne nulle et de variance $\sigma_{\theta}^2 = 0.1^2$. La matrice de transition Q_{α} est fixée à :

$$Q_{\alpha} = \begin{bmatrix} 0.6 & 0.25 & 0.1 & 0.05 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.1 & 0.2 & 0.5 & 0.2 \\ 0.05 & 0.1 & 0.25 & 0.6 \end{bmatrix}$$
(5.20)

Comme la matrice définie dans l'équation 5.18, la matrice définie ici permet de facilement changer de position dans l'ordre de traitement, afin d'être robuste aux changements de visibilité des doigts. La difficulté de cette application est que les doigts peuvent être



FIG. 5.13 – Tests sur la log-vraisemblance utilisée dans la séquence de fourmis. (a) Image de test, dans laquelle nous souhaitons trouver les fourmis désignées en vert. Résultats obtenus en utilisant comme orientation celle désignée par le rectangle jaune (superposé à l'image de vraisemblance) avec un angle (en radian) de (b) 0, (c) 0.39, (d) 0.79, (e) 1.18, (f) 1.57, (g) 1.96, (h) 2.36, et (i) 2.75. Une couleur rouge indique une forte vraisemblance.



(c)

FIG. 5.14 – Résultats de suivi de fourmis. (a) RPS sans constrainte spatiale (les flèches bleues indiquent les fourmis qui ne sont plus suivies), (b) PS avec constrainte spatiale et (c) RPS avec constrainte spatiale. Toutes les fourmis sont bien suivies dans les séquences (b) et (c).

partiellement ou totalement cachés. Nous souhaitons alors suivre la main tout en gardant une cohérence globale de la forme, en utilisant des contraintes spatiales floues. Bien qu'elles puissent être automatiquement apprises, les relations spatiales entre les doigts sont fixées. Nous avons mis en œuvre quatre contraintes spatiales : distance, angle, alignement et exclusion, en prenant comme opérateur de fusion Ξ entre les contraintes l'opérateur min (toutes les contraintes doivent être satisfaites). Nous avons défini :

- deux valeurs d'angle, avec une imprécision caractérisée par un patron trapézoïdal flou de longueur de support $\pi/4$:
 - \circ environ $\pi/8$ et
 - environ $-\pi/8$;
- deux valeurs de distance :
 - *proche* et
 - \circ loin;
- une contrainte d'alignement : en utilisant une régression linéaire, les doigts ne peuvent s'écarter de celle-ci d'un certaine distance seuil fixée ;
- une contrainte d'exclusion : la superposition de deux doigts est interdite dès qu'elle dépasse 50% de leurs aires totales.

Donnons un exemple de formulation des contraintes spatiales floues en considérant l'état du *majeur* \mathbf{x}_t^m , et en supposant que les autres doigts (notés \mathbf{x}_t^i , \mathbf{x}_t^{an} et \mathbf{x}_t^{au} respectivement pour *index*, *annulaire* et *auriculaire*) ont déjà été traités dans le scénario :

$$\begin{split} \mu_{\tilde{\mathbf{x}}_{t}}(\mathbf{x}_{t}^{m}) &= \nu_{\tilde{\mathbf{x}}_{t}}(\mathbf{x}_{t}^{m})^{\gamma} \\ \nu_{\tilde{\mathbf{x}}_{t}}(\mathbf{x}_{t}^{m}) &= \frac{K}{k=1}\nu_{\tilde{\mathbf{x}}_{t}}^{k}(\mathbf{x}_{t}^{m}) \\ &= \min\left(^{B}\nu_{\tilde{\mathbf{x}}_{t}}^{distance}(\mathbf{x}_{t}^{m}), ^{B}\nu_{\tilde{\mathbf{x}}_{t}}^{angle}(\mathbf{x}_{t}^{m}), \\ & Q_{u}^{alignement}(\mathbf{x}_{t}^{m}), ^{B}\nu_{\tilde{\mathbf{x}}_{t}}^{exclusion}(\mathbf{x}_{t}^{m})\right) \\ ^{B}\nu_{\tilde{\mathbf{x}}_{t}}^{distance}(\mathbf{x}_{t}^{m}) &= \frac{L}{\psi}^{B}\nu_{\tilde{\mathbf{x}}_{t}}^{distance}(\mathbf{x}_{t}^{m}) \\ &= \min\left(^{B}\nu_{\tilde{\mathbf{x}}_{a}^{an}}^{distance \ est \ proche \ de}(\mathbf{x}_{t}^{m}), \\ & B_{v}_{\tilde{\mathbf{x}}_{t}}^{distance \ est \ proche \ de}(\mathbf{x}_{t}^{m}), \\ & B_{v}_{\tilde{\mathbf{x}}_{t}}^{distance \ est \ proche \ de}(\mathbf{x}_{t}^{m}), \\ & B_{v}_{\tilde{\mathbf{x}}_{t}}^{distance \ est \ proche \ de}(\mathbf{x}_{t}^{m}), \\ & B_{v}_{\tilde{\mathbf{x}}_{t}}^{distance \ est \ proche \ de}(\mathbf{x}_{t}^{m}), \\ & B_{v}_{\tilde{\mathbf{x}}_{t}}^{distance \ est \ proche \ de}(\mathbf{x}_{t}^{m}), \\ & B_{v}_{\tilde{\mathbf{x}}_{t}}^{distance \ est \ proche \ de}(\mathbf{x}_{t}^{m}), \\ & B_{v}_{\tilde{\mathbf{x}}_{t}}^{distance \ est \ proche \ de}(\mathbf{x}_{t}^{m}), \\ & B_{v}_{\tilde{\mathbf{x}}_{t}}^{distance \ est \ proche \ de}(\mathbf{x}_{t}^{m}), \\ & B_{v}_{\tilde{\mathbf{x}}_{t}}^{angle \ model{stance} \ est \ loin \ de}(\mathbf{x}_{t}^{m}), \\ & B_{v}_{\tilde{\mathbf{x}}_{t}}^{angle \ model{stance} \ est \ loin \ de}(\mathbf{x}_{t}^{m}), \\ & B_{v}_{\tilde{\mathbf{x}}_{t}}^{angle \ model{stance} \ model{stan$$

avec des fonctions d'appartenance :

- binaires :
 - de distance, qui déterminent si un objet est proche ou éloigné d'un autre, selon un seuil fixé par la distance entre deux doigts consécutifs, et la distance de la main à l'appareil de mesure;
 - d'angle, qui déterminent avec quel degré un angle mesuré entre deux doigts satisfait la valeur jugée idéale (ici, $\pi/8$ et $-\pi/8$);
 - d'exclusion, qui déterminent avec quel degré deux objets ne se chevauchent pas ;
- ternaires (ou quaternaires, cela dépend du nombre d'objets déjà traités) :
 - d'alignement, qui déterminent avec quel degré trois doigts sont alignés, en utilisant une régression linéaire calculée à partir de la position de la base d'au moins deux doigts déjà traités, et dont la fonction d'appartenance définie sur la distance orthogonale du doigt considéré à la droite est décroissante.

Notons que des optimisations de calcul sont possibles. Dans l'exemple donné ici, le calcul de l'exclusion entre le majeur et l'index est par exemple inutile, tout comme l'orientation, puisque l'annulaire a déjà satisfait, auparavant, les contraintes imposées par l'index (ou l'inverse, ce qui revient au même). Afin d'illustrer les contraintes spatiales obtenues dans le domaine de l'image, nous choisissons une image représentative de la séquence étudiée (figure 5.15).



FIG. 5.15 – Image tirée de la séquence, sur laquelle nous souhaitons contraindre la position spatiale et angulaire du *majeur*, sachant que les états correspondant aux doigts *index*, *annulaire* et *auriculaire* ont déjà été traités.

La contrainte spatiale de distance est illustrée sur la figure 5.16. Une valeur sombre indique que la contrainte est satisfaite. L'objectif de la relation de distance est de contraindre la localisation du point central du doigt *majeur*. En utilisant les contraintes formulées par les doigts *index* (figure 5.16(a)), *annulaire* (figure 5.16(b)), et *auriculaire* (figure 5.16(c)), la fusion obtenue dans le domaine de l'image est représentée sur la figure 5.16(d). Rappelons toutefois que la présentation suggérée ici permet juste d'illustrer les contraintes, leurs calculs ne nécessitent pas d'être projetés dans le domaine de l'image (sauf pour le principe d'exclusion), ce qui signifie qu'elles ne sont pas évaluées sur l'ensemble des pixels de l'image, mais seulement sur une réalisation donnée de l'état du doigt *majeur* x_t^m .



FIG. 5.16 – Contraintes spatiales floues du concept de distance dans le cas de la main, par rapport au centre du doigt *majeur*. (a) Valeur *est proche de* de l'*index*, (b) valeur *est proche de* de l'*annulaire*, (c) valeur *est éloigné de* de l'*auriculaire* et (d) résultat global contraint en utilisant un opérateur de fusion $\psi = \min$.

La contrainte spatiale d'angle est illustrée sur la figure 5.17. Cette contrainte restreint les possibilités d'orientation de l'objet. La fusion opérée dans l'image 5.17(d) est effectuée dans le domaine des orientations, et non le domaine spatial, les images proposées ne sont donc que des projections des orientations à des positions fixées (respectivement, pour les images 5.17(a), 5.17(b), 5.17(c), et 5.17(d), aux positions centrales des doigts *index*, *annulaire*, *auriculaire*, et *majeur*).

La contrainte spatiale d'exclusion est illustrée sur la figure 5.18. Il est nécessaire, afin d'illustrer cette contrainte, de fixer l'orientation, puisque la contrainte dépend à la fois de l'orientation et de la localisation centrale de l'objet. L'orientation considérée est donc représentée par un état hypothétique de couleur bleue superposé aux images de contrainte.

La contrainte spatiale d'alignement est illustrée sur la figure 5.19. Le résultat est obtenu directement, sans processus de fusion, puisqu'il s'agit d'un opérateur quaternaire. Un opérateur ternaire aurait été utilisé si seules les positions de deux objets avaient été connues. Lorsque moins de deux objets ont déjà été traités, cette contrainte est ignorée.

Le résultat du processus de fusion pour la localisation centrale de l'objet des contraintes de distance, d'exclusion et d'alignement, en prenant une orientation fixée, est présenté sur la figure 5.20.

Il aurait été possible de définir d'autres contraintes spatiales, plus évoluées ou plus robustes, mais celles définies ici permettent de représenter un grand nombre de configurations impossibles, tout en restant simples à calculer. Par exemple, nous aurions pu utiliser une notion d'alignement plus fine [Vanegas et al., 2010], et au lieu d'utiliser un principe d'exclusion, modéliser le concept « traverser » aurait également semblé judicieux [Vanegas et al., 2009].

La vraisemblance utilise les valeurs de gradient maximales sur les lignes normales des points de la B-spline (c'est une version un peu simplifiée de celle proposée dans [Isard et



FIG. 5.17 – Contraintes spatiales floues du concept d'angle dans le cas de la main, par rapport à l'orientation du doigt *majeur*. (a) Valeur $-\pi/8$ de l'*index*, (b) valeur $\pi/8$ de l'*annulaire*, (c) valeur $\pi/8$ de l'*auriculaire* et (d) résultat global contraint en utilisant un opérateur de fusion $\psi = \min$.



FIG. 5.18 – Contraintes spatiales floues du concept d'exclusion dans le cas de la main, par rapport à la position et l'orientation du doigt *majeur*. Nous considérons l'orientation du doigt dont la représentation est superposée en couleur jaune aux images de contrainte. Principes d'exclusion vus par (a) l'*index*, (b) l'*annulaire*, (c) l'*auriculaire* et (d) exclusion globale obtenue en utilisant un opérateur de fusion $\psi = \min$.



FIG. 5.19 – Contraintes spatiales floues du concept d'alignement dans le cas de la main, par rapport à la position du doigt *majeur*. Les positions des doigts déjà traités *index*, *annulaire* et *auriculaire* servent à construire une droite floue, par régression linéaire, et contraindre ainsi la position du *majeur*.



FIG. 5.20 – Fusion des contraintes spatiales globales de position obtenues par les concepts (a) de distance, (b) d'exclusion et (c) d'alignement, en considérant l'orientation du doigt dont la représentation est superposée aux images de contrainte d'exclusion (en couleur jaune) et de résultat final (en couleur bleue). Le résultat de la fusion pour la position du doigt *majeur* est illustré sur la figure (d).

Blake, 1998a]):

$$p(\mathbf{y}_t | \mathbf{x}_t^i) \propto \exp\left(\frac{\lambda}{P_{a^i}} \sum_{p=1}^{P_{a^i}} \max_{x \in \overline{n_{a^i_p}}} \nabla\left[I_t(x) \star \mathcal{N}(x, \sigma^2)\right]\right)$$
(5.21)

où ∇ et * sont respectivement les opérateurs de gradient et de convolution, a_p^i le *p*-ième point de la B-Spline du doigt *i* de longueur P_{a^i} et défini par \mathbf{x}_t^i , $\overrightarrow{n_{a_p^i}}$ le segment normal au point a_p^i , $I_t(x)$ la valeur du niveau de gris au point *x* dans l'image I_t au temps *t*, $\lambda = 30$ une valeur constante multiplicative fixée empiriquement, et $\mathcal{N}(x, \sigma^2)$ une distribution normale de moyenne *x* et de variance fixée $\sigma^2 = 0.1$ afin de rendre l'observation un peu plus robuste au bruit. La figure 5.21 présente les résultats obtenus par la vraisemblance. Dans la mesure où nous souhaitons estimer la position des doigts ainsi que leur orientation, il est nécessaire de construire une vraisemblance robuste à cette dernière. Nous pouvons constater que les réponses obtenues sont également fortes pour les autres doigts, d'où la nécessité de contraindre le suivi, afin que tous les doigts estimés ne se localisent pas sur les (mêmes) maxima locaux de la vraisemblance.



FIG. 5.21 – Tests sur la log-vraisemblance utilisée dans la séquence de main. (a) Image de test, dans laquelle nous souhaitons trouver le majeur désigné en vert, résultats obtenus en utilisant la forme du majeur représentée en jaune avec un angle (en radian, et sachant que la bonne valeur est -0.07) de (b) 0, (c) 0.39, (d) 0.79, (e) 1.18, (f) 1.57, (g) 1.96, (h) 2.36, et (i) 2.75. Une couleur rouge indique une forte vraisemblance.

Les erreurs de forme des doigts *index, majeur, annulaire* et *auriculaire* avec les méthodes RPS sans contrainte spatiale, PS avec contraintes spatiales et RPS avec contraintes spatiales sont données sur la figure 5.22, avec N = 2000 particules. Comme dans la section 4.4, les erreurs de forme sont égales à 1 moins le taux de recouvrement. Ce taux correspond au ratio entre les points communs de la forme estimée et la vraie forme (qui a été définie manuellement) et le maximum des aires de ces deux objets. Les résultats en utilisant une stratégie d'échantillonnage partitionné simple ont été générés avec un ordre de traitement aléatoire (dans cette séquence particulière, un ordre fixé pouvait donner des résultats moins bons, puisque certains doigts sont fréquemment partiellement cachés, ce qui fait chuter leur vraisemblance).

Les erreurs obtenues par le RPS avec contraintes spatiales sur la figure 5.22(c) sont plus faibles que celles obtenues par les deux autres méthodes (figures 5.22(a) et 5.22(b)), et n'atteignent jamais plus de 36% de mauvais recouvrement des formes, tous doigts confondus. Les résultats des formes correspondantes sont illustrés sur la figure 5.23 et ont été obtenus avec N = 2000 particules. Le RPS sans contrainte spatiale échoue dès qu'un doigt est caché, ou lorsque les doigts sont rapprochés et que cela crée une ambiguité, c'est-à-dire, plusieurs modes, dans la densité de vraisemblance (à partir de la troisième image sur la figure 5.23(a) ($t \sim 400$)). Le RPS donne des résultats meilleurs que ceux du PS grâce à l'estimation du scénario, ce qui permet d'estimer en premier les doigts ayant une vraisemblance forte, c'est-à-dire ceux qui sont visibles, puis les autres, qui sont plus contraints par les relations spatiales floues (à partir de $t \sim 500$ sur les figures 5.23(b), 5.23(c), et 5.22(b), 5.22(c)). Cela est particulièrement évident à la fin de la séquence ($t \sim 770$), où le doigt *index* avec la méthode PS est mal estimé, et prend la place du majeur. Ainsi, par les contraintes spatiales floues, cela conditionne l'estimation du majeur, qui se retrouve bloqué entre le vrai *majeur* et l'annulaire (dernière image sur la figure 5.23(b), et figure 5.22(b)).

Les résultats des tests que nous avons obtenus sur le PS et le RPS avec contraintes spatiales sont assez variables, de par la nature stochastique de l'algorithme, et la difficulté en termes de simulation de la séquence (quatre objets, et de fortes contraintes sur leur disposition spatiale). Ainsi, nous avons choisi de présenter ici des résultats de suivi classiques, ce qui signifie que les méthodes peuvent obtenir de meilleurs, comme de moins bons, résultats. Nous observons toutefois une supériorité quasi systématique du RPS par rapport au PS.

5.6 Conclusion

Nous avons présenté dans ce chapitre deux contributions. Premièrement, nous avons introduit des contraintes spatiales floues dans un filtre particulaire pour le suivi multiobjets. Cette nouvelle information permet de facilement gérer des contraintes de positionnement entre objets, telles que *est à droite de, est derrière, est aligné avec*, etc. En modélisant des opérateurs flous unaires, binaires, et plus généralement *n*-aires, cela permet d'évaluer la possibilité d'une configuration d'objets. Cette possibilité, traduite par un ensemble flou désignant avec quel degré un ensemble de contraintes est satisfait, est ensuite injectée dans un formalisme probabiliste afin de juger de la probabilité d'une configuration d'objets à un instant *t*.

Comme deuxième contribution, le filtre particulaire multi-objets proposé utilise une stratégie d'échantillonnage partitionné à rang, ce qui, comme l'échantillonnage partitionné [MacCormick et Blake, 2000], atténue le problème de la dimension en procédant de manière séquentielle à des étapes de rééchantillonnages pondérés dans les espaces d'états des objets pris individuellement. Mais contrairement à la méthode classique d'échantillonnage partitionnage partitionné partitionné, l'ordre de traitement proposé dans notre approche est également estimé par le filtre particulaire, ce qui rend le suivi plus robuste et diminue le phénomène



FIG. 5.22 – Erreurs de forme du suivi de la main. (a) RPS sans contrainte spatiale, (b) PS avec contraintes spatiales et (c) RPS avec contraintes spatiales.

d'appauvrissement, tout en gardant un temps de calcul identique à celui de l'échantillonnage partitionné.

Toutefois, un certain nombre d'améliorations pourraient être apportées au modèle présenté. Premièrement, bien que le fait d'ajouter des contraintes au moment du calcul de la vraisemblance et non pendant la simulation soit un choix généralement pratiqué dans la littérature, une conséquence immédiate est que, de la même manière qu'avec une vraisemblance trop « piquée », ajouter des contraintes au moment de la mise à jour des poids s'expose au risque d'obtenir des poids de valeurs trop faibles, en raison de la présence d'un certain nombre de contraintes trop fortes. Une solution simple consiste à relâcher légèrement les contraintes, afin de ne pas faire diverger le filtre. Mais bien évidemment, lorsque cela est possible, la meilleure solution est de générer les états conditionnellement aux contraintes. Cette solution a été prévue dans le modèle proposé (équation 5.13), mais non considérée dans nos tests car elle est réalisable uniquement lorsque les contraintes sont simples, ce qui n'était pas l'objet de notre contribution.

La modélisation de contraintes spatiales floues pourrait également être plus avancée, en manipulant des informations plus complexes, fréquemment utilisées dans une problématique de description de scène [Colliot, 2003; Colliot et al., 2006; Nempont et al., 2008; Nempont, 2009; Deruyver et Hodé, 1997; Deruyver et al., 2005]. Nous pourrions également imaginer apprendre les paramètres des contraintes spatiales floues automatiquement, comme cela a déjà été étudié dans la littérature [Atif et al., 2007; Hudelot et al., 2008].

Bien qu'elle n'ait pas été observée en pratique, une autre limite mérite d'être signalée. Cela concerne l'algorithme de l'échantillonnage partitionné à rang, qui, en simulant en premier lieu l'ordre de traitements des objets, puis en rééchantillonnant *M* fois le nuage de particules, perd systématiquement de la diversité au niveau des scénarios, ce qui peut certainement être un problème lorsque le nombre d'objets considéré est grand. Une autre



FIG. 5.23 – Résultats du suivi de la main, aux temps 100, 258, 432, 529, 620, 685, 740 et 770. (a) RPS sans contrainte spatiale conduisant à des échecs sur plusieurs images (indiqués par des flèches), (b) PS avec contraintes spatiales et (c) RPS avec contraintes spatiales.

perspective à ce travail serait de comparer les résultats obtenus à ceux du DPS (pour *Dynamic Partitioned Sampling*) proposé par [Smith et Gatica-Perez, 2004].

Enfin, une dernière amélioration du filtre multi-objets proposé dans ce chapitre serait de considérer un nombre d'objets M aléatoire, comme cela a par exemple été effectué dans [Hue et al., 2002; Smith et al., 2005].

Chapitre 6

Vraisemblances à modèles multiples

Sommaire

6.1	Travaux existants							
	6.1.1	Multi-modalités						
		6.1.1.1 Probabilité de pertinence d'une modalité						
		6.1.1.2 Confiance dans une modalité						
	6.1.2	Apparence changeante						
	6.1.3	Discussion						
6.2	Vraise	Vraisemblance à modèles multiples						
	6.2.1	Travaux existants						
	6.2.2	Modèle proposé						
	6.2.3	Applications						
6.3	Machine à vecteurs de support (SVM) 174							
	6.3.1	Principe du SVM à deux classes						
	6.3.2	Extensions à plusieurs classes						
		6.3.2.1 Un contre tous (WTA)						
		6.3.2.2 Vainqueur par maximum de victoires (MWS) 177						
		6.3.2.3 Stratégie <i>Pairwise Coupling</i> 177						
	6.3.3	Intégration dans une vraisemblance à modèles multiples 177						
6.4	Expér	Expériences						
	6.4.1	Protocole de tests						
		6.4.1.1 Présentation de la séquence						
		6.4.1.2 Modèle dynamique						
		6.4.1.3 Modèle de vraisemblance						
	6.4.2	Apprentissage SVM181						
		6.4.2.1 Critères de distance entre histogrammes de couleurs 181						
		6.4.2.2 Critères de distance euclidienne entre formes 183						
		6.4.2.3 Critères de distance de Hausdorff entre formes 183						
	6.4.3	Résultats préliminaires						
6.5	Concl	Conclusion						

L'objet de ce chapitre est d'étudier l'intégration de plusieurs modalités dans le filtre particulaire. Le terme de multi-modalités désigne, au sens large, l'utilisation de plusieurs caractéristiques issues d'un capteur, telles qu'une information de contours et de couleurs

dans une image, ou de plusieurs capteurs, tels qu'une caméra et un laser, ou simplement plusieurs caméras de même type.

L'objectif des approches utilisant plusieurs modalités est de définir un modèle intégrant les différentes sources d'observation d'un objet afin d'en rendre le suivi plus robuste. Cela sous-entend d'utiliser, si possible, les spécificités des capteurs, leurs complémentarités comme leurs redondances, et de détecter si l'information délivrée est pertinente ou non.

Ce dernier point évoque la problématique des filtres modélisant les fausses alarmes, qui sont des informations erronées délivrées par le capteur (ou un processus de détection lié au capteur), généralement à cause d'une défaillance de celui-ci ou d'un environnement bruité. Cependant, la manière de faire est différente : les modèles avec fausses alarmes formalisent mathématiquement la probabilité qu'une observation soit une fausse alarme, et l'introduisent ensuite dans le modèle de mesure, alors que les modèles multi-modaux tentent généralement de définir et d'estimer une notion de confiance, ou d'incertitude, sur les observations.

Nous présenterons tout d'abord les traits communs des modèles utilisant plusieurs modalités (section 6.1), avant de passer à la contribution de ce chapitre, qui propose de définir un modèle de vraisemblance à plusieurs composantes (section 6.2). Cela permet par exemple de gérer le cas où l'apparence de l'objet évolue au cours du temps, en estimant automatiquement les poids de la densité de probabilité de mélange de la vraisemblance, dont chaque composante représente une modalité d'observation. Les poids peuvent être calculés par un classifieur, comme nous le verrons dans la section 6.3. La section 6.4 propose enfin une série d'expériences sur la problématique du suivi de bouche.

Ce travail a été réalisé avec la collaboration d'un ancien doctorant de Télécom Paris-Tech, devenu ATER au laboratoire d'informatique de Paris 6, Emi Aldéa. Les contributions et les résultats proposés dans ce chapitre n'en sont qu'à un stade préliminaire.

6.1 Travaux existants

6.1.1 Multi-modalités

Utiliser plusieurs caractéristiques, plusieurs capteurs, ou plusieurs modèles d'apparence, sont des concepts bien distincts. Le premier signifie qu'un capteur fournit plusieurs caractéristiques, telles que des caractéristiques de texture, de forme, de couleur. Plusieurs capteurs indiquent que plusieurs dispositifs délivrant des observations, synchrones ou non, sont intégrées dans le filtre particulaire. Enfin, utiliser plusieurs modèles d'apparence permet de représenter précisément le modèle de référence d'un objet, et de conditionner cette information à la vraisemblance d'une caractéristique. La distinction entre modèles multi-capteurs et modèles multi-caractéristiques n'est pas toujours faite dans la littérature. En effet, dans le cadre du suivi par filtrage particulaire, ces problématiques possèdent des points communs, si bien que leurs mises en œuvre sont parfois identiques, et ne font pas de différence entre ces deux types d'information. Plutôt qu'une énumération exhaustive, nous proposons de décrire deux des types de modèles les plus populaires, qui conviennent dans la plupart des situations, en permettant notamment de ne pas faire la distinction entre caractéristique et capteur (sections 6.1.1.1 et 6.1.1.2). Ainsi, nous utilisons le terme multi-modalités pour désigner à la fois multi-caractéristiques et multi-capteurs. Rappelons enfin qu'une introduction plus générale de l'utilisation de plusieurs modalités a été présentée dans la section 1.3.1.3.

La variable aléatoire $\mathbf{x}_t \in \mathcal{X}$ désigne l'état caché (d'un objet) au temps t. Dans cette section, les observations sont regroupées dans un vecteur de R composantes, $\mathbf{y}_t = (\mathbf{y}_t^1, \dots, \mathbf{y}_t^R)$ avec \mathbf{y}_t^r la r-ième modalité.

6.1.1.1 Probabilité de pertinence d'une modalité

La première méthode consiste à décomposer la densité de probabilité de la vraisemblance sous forme d'une densité de probabilité de mélange, dont chaque composante représente une modalité. Nous notons l la variable aléatoire à valeurs dans $\{1, ..., R\}$ indiquant l'indice de la modalité considérée :

$$p(\mathbf{y}_t|\mathbf{x}_t) = \sum_{r=1}^{R} p(\mathbf{l} = r|\mathbf{x}_t) p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{l} = r)$$
$$= \sum_{r=1}^{R} \pi_t^r p(\mathbf{y}_t^r|\mathbf{x}_t)$$
(6.1)

où $\sum_{r=1}^{R} \pi_t^r = 1$, avec $\{\pi_t^r\}_{r=1}^{R}$ les probabilités de pertinence des modalités. Sémantiquement, π_t^r représente donc la probabilité que la *r*-ième modalité soit celle qui décrive l'état \mathbf{x}_t . Elles sont soit fixées à la main [Xu et Li, 2005], soit adaptatives mais avec un ensemble des valeurs possibles fini et fixé [Hotta, 2006]. Le principal problème de ce modèle est la difficulté de définir des poids $\{\pi_t^r\}_{r=1}^R$ adaptatifs, puisque, par nature, ils sont dépendants entre eux. Le modèle d'observations à sauts proposé dans [Caron et al., 2007] se rapproche de celui présenté dans l'équation 6.1, mais nous reviendrons sur ce point dans la section 6.2.

6.1.1.2 Confiance dans une modalité

Le modèle le plus fréquemment utilisé est sans doute celui que nous présentons maintenant. Il traduit une notion de confiance, ou de fiabilité, dans la modalité considérée. Ce modèle considère tout d'abord une indépendance conditionnelle entre les modalités sachant l'état \mathbf{x}_t :

$$p(\mathbf{y}_t|\mathbf{x}_t) = \prod_{r=1}^R p(\mathbf{y}_t^r|\mathbf{x}_t)$$
(6.2)

Ensuite, sont ajoutés de manière *ad hoc*, comme exposants des vraisemblances marginales $p(\mathbf{y}_t^r | \mathbf{x}_t), r = 1, ..., R$, les indices de confiance $\{\alpha_t^r\}_{r=1}^R$ tels que $\forall r \in \{1, ..., R\}, \alpha_t^r \in [0, 1]$:

$$p(\mathbf{y}_t|\mathbf{x}_t) = \prod_{r=1}^R p(\mathbf{y}_t^r|\mathbf{x}_t)^{\alpha_t^r}$$
(6.3)

Ainsi α_t^r indique la confiance dans la modalité r. Contrairement au modèle présenté dans l'équation 6.1, nous pouvons remarquer l'indépendance des indices α_t^r entre eux, ce qui facilite leur mise à jour. Par exemple, dans [Brasnett et al., 2007], les indices $\tilde{\alpha}_t^r$ au temps t sont calculés selon :

$$\tilde{\alpha}_t^r = \frac{1}{d_{r,\min}^2} \tag{6.4}$$

avec $d_{r,\min}^2$ la distance minimale obtenue entre le modèle de référence désignant la modalité r et tous les candidats désignés par les particules. Ainsi, si la modalité r désigne une caractéristique de couleur, la distance d_r^2 utilisée pourrait être celle de Bhattacharyya, entre un modèle de référence d'histogramme de couleurs, et les histogrammes candidats définis par le nuage de particules. Donc, moins la distance entre le patron connu et la particule gagnante est élevée, plus la quantité $\tilde{\alpha}_t^r$ sera forte. Cependant, cette quantité n'est pas définie dans [0, 1], et ne représente donc pas un indice de confiance. Les auteurs proposent alors de normaliser les poids, $\alpha_t^r = \tilde{\alpha}_t^r / \sum_{u=1}^R \tilde{\alpha}_t^u$, ce qui induit une dépendance, en obtenant donc une notion de confiance relative. Dans [Erdem et al., 2010], les indices de confiance sont mis à jour selon :

$$\alpha_t^r = \alpha_{t-1}^r + \nu (s_t^r - \alpha_{t-1}^r)$$
(6.5)

avec ν un facteur de mise à jour fixé à 0.1, et s_t^r une mesure de qualité définie par une notion d'intégration démocratique, proposée dans [Triesch et Malsburg, 2001], qui consiste à avantager une modalité qui fournit un résultat en accord avec celui proposé par l'ensemble des modalités. Elle s'écrit :

$$s_{t}^{r} = \begin{cases} p(\mathbf{y}_{t}^{r}|\hat{\mathbf{x}}_{t}) - \frac{1}{N} \sum_{n=1}^{N} p(\mathbf{y}_{t}^{r}|\mathbf{x}_{t}^{(n)}) & \text{si} \quad p(\mathbf{y}_{t}^{r}|\hat{\mathbf{x}}_{t}) > \frac{1}{N} \sum_{n=1}^{N} p(\mathbf{y}_{t}^{r}|\mathbf{x}_{t}^{(n)}) \\ 0 & \text{sinon} \end{cases}$$
(6.6)

avec $p(\mathbf{y}_t^r | \hat{\mathbf{x}}_t)$ la vraisemblance associée à la modalité r, conditionnellement au candidat $\hat{\mathbf{x}}_t$ estimé en prenant en compte toutes les modalités, et $\sum_{n=1}^N p(\mathbf{y}_t^r | \mathbf{x}_t^{(n)})/N$ la valeur de vraisemblance moyenne associée à la modalité r selon toutes les particules $\{\mathbf{x}_t^{(n)}\}_{n=1}^N$. Ainsi, si la vraisemblance de la modalité r conditionnelle au candidat estimé est supérieure à la vraisemblance moyenne, cela signifie que l'estimé global est cohérent vis-à-vis de cette modalité, et sa confiance est ainsi accrue.

Notons que mettre en exposant de la vraisemblance les indices de confiance (équation 6.3) n'est mathématiquement pas rigoureux, car, à cause des indices $\{\alpha_t^r\}_{r=1}^R$, la vraisemblance jointe n'est plus directement issue des hypothèses d'indépendance conditionnelles, qui se traduisent par une décomposition en facteur des vraisemblances marginales $p(\mathbf{y}_t^r|\mathbf{x}_t)$, ce qui rend difficile son interprétation. De plus, la densité jointe ne s'intègre généralement plus à 1, il manque donc un terme de normalisation. La justification de ce choix n'est pas effectuée par les auteurs dans [Brasnett et al., 2007; Erdem et al., 2010], mais nous pouvons considérer qu'il est d'ordre heuristique, et permet de simplement retranscrire l'idée intuitive qu'il y a derrière.

6.1.2 Apparence changeante

Lorsque l'apparence de l'objet évolue dans le temps, par exemple à cause de changements de luminosité, les algorithmes de suivi utilisant un critère de corrélation entre un modèle de référence et un candidat doivent remettre à jour le modèle, afin de rester robuste aux changements. La mise en œuvre d'un modèle avec une apparence changeante consiste donc à mettre à jour progressivement le modèle de référence, ce qui se distingue de celle que nous étudions, qui consiste à modéliser explicitement plusieurs composantes, qui peuvent être liées à des apparences. Toutefois, ces deux types d'approches ont une problématique commune, c'est pourquoi nous faisons un bref état de l'art à ce sujet. Dans la littérature sur le filtre particulaire, ce problème n'est que rarement considéré. Dans [Nummiaro et al., 2002; Muñoz-Salinas et al., 2008a], l'histogramme de référence h_t^{\bullet} au temps t est mis à jour selon :

$$h_t^{\bullet} = (1 - \gamma) h_{t-1}^{\bullet} + \gamma \hat{h}_t$$
(6.7)

avec $\gamma \in [0, 1]$ un terme contrôlant la mise à jour et \hat{h}_t un histogramme candidat. Celuici correspond à l'histogramme obtenu par l'estimé du filtre particulaire, et l'étape de mise à jour n'est effectuée que lorsque sa vraisemblance est supérieure à un seuil fixé. Dans [Nummiaro et al., 2002], la quantité γ est fixée, alors que dans [Muñoz-Salinas et al., 2008a], elle dépend de la similarité entre l'histogramme de référence h_{t-1}^{\bullet} et celui qui est défini par l'estimé \hat{h}_t , de sorte que γ est d'autant plus élevé que la différence entre ces deux histogrammes est grande.

6.1.3 Discussion

Même si l'objectif de notre approche est de définir une vraisemblance à modèles multiples, ce qui se différencie des contextes multi-caractéristiques et multi-capteurs, et légèrement du contexte d'apparence changeante, nous pouvons en tirer quelques enseignements. Toutes les méthodes présentées dans cette section visent à définir des poids, qu'ils soient de probabilité, de confiance, ou de mise à jour de modèle, adaptatifs. C'est ce caractère adaptatif qui rend un modèle à la fois souple et robuste vis-à-vis du contexte. Cependant, il est souvent difficile de mettre à jour ces quantités, et cela est effectué de manière plutôt heuristique, en calculant *a posteriori* les valeurs selon un critère défini. La stratégie n'est donc pas intégrée au filtre particulaire, et délivre un jeu de paramètres unique pour l'ensemble des particules. Cela peut s'avérer dangereux, car en mettant mal à jour les paramètres du modèle, la vraisemblance ainsi considérée devient obsolète, ce qui, immédiatement, a de fortes chances de pénaliser l'algorithme de suivi en le faisant diverger. Ainsi, en accumulant des erreurs au cours du temps, cela détériore, souvent irrémédiablement, le modèle de réference mis à jour.

Le modèle que nous proposons définit la vraisemblance par une densité de probabilité de mélange, dont les poids sont attachés aux particules. Un poids différent étant calculé pour chaque particule, le problème que nous venons de soulever n'existe plus. L'originalité vient du fait qu'un poids est lié, non pas à une caractéristique ou un capteur, mais à une décomposition de l'état et de l'observation. Cela suppose que la fonction de décision de la décomposition caractérisant le poids soit établie au préalable.

6.2 Vraisemblance à modèles multiples

Nous proposons dans cette section de définir une vraisemblance à modèles multiples. Pour simplifier, considérons dans un premier temps que chaque modèle décrit une apparence de l'objet. Une apparence est une représentation possible de l'objet, selon une caractéristique donnée. Cette modélisation est adaptée lorsqu'un objet change d'apparence (couleur, forme,...) au cours de la séquence ou lorsque les conditions de luminosité sont changeantes. Par exemple, dans le contexte du suivi de visage en 3*D*, nous pourrions définir plusieurs composantes, appelées postures, dont les probabilités sont calculées en utilisant un critère d'orientation de la tête (*face, profil* et *derrière*), et en la couplant avec une vraisemblance de couleurs conditionnée par la posture considérée. L'apparence correspond au modèle de référence qui conditionne la vraisemblance. Dans ce cas, le critère d'orientation aide à définir le type de posture que l'état de l'objet inconnu décrit. En définissant la vraisemblance jointe par une densité de probabilité de mélange, les poids de chacune des composantes sont définis en utilisant l'orientation de la tête (estimée par le filtre particulaire). Ainsi, cela permet d'intégrer de manière originale des caractéristiques complémentaires dans la fonction de vraisemblance jointe. L'approche proposée est résumée sur la figure 6.1.



FIG. 6.1 – Résumé de l'approche proposée. L'apprentissage des modèles peut être fait de manière automatique, grâce à une base d'apprentissage, ou fixé à la main. Dans ce chapitre, nous considérons un apprentissage par SVM, que nous étudierons dans la section 6.3. Celui-ci sert à décomposer l'objet, ici la bouche, en plusieurs classes, que nous appelons postures. Dans cet exemple, les points de contrôle en bleu sont utilisés pour définir le critère de séparation du SVM. La base d'apprentissage ainsi que le classifieur SVM permettent de déterminer trois modèles d'apparence, qui, dans cet exemple, utilisent une information de couleur (étape d'extraction de trois modèles d'apparence). Dans l'algorithme de suivi, lorsqu'une bouche candidate est proposée, la vraisemblance totale correspond à la pondération des vraisemblances marginales de chacune des apparences par les poids d'appartenance de la forme aux postures considérées.

6.2.1 Travaux existants

Cette problématique n'a, à notre connaissance, jamais été présentée sous cette forme. Les travaux se rapprochant le plus de notre proposition sont sans doute ceux proposés dans [Caron et al., 2007]. Cette approche consiste à définir des modèles d'observations à sauts. Ainsi, un capteur peut se trouver dans plusieurs états de fonctionnement, chacun étant lié à un modèle de vraisemblance. Le modèle à sauts est défini dans le filtre particulaire, en ajoutant à l'estimation de x_t l'indice du régime dans lequel le capteur se trouve. La vraisemblance est ainsi définie par l'indice du régime, et évidemment, x_t . L'application traitée par les auteurs concerne l'estimation de la position d'un véhicule, avec des capteurs tels qu'un GPS, une centrale inertielle, une boussole, une caméra, etc., et pour lesquels ils définissent un ou deux modèles de vraisemblance selon que les capteurs délivrent une information correcte ou défaillante. Le modèle à sauts qu'ils mettent en place peut également s'interpréter par une densité de mélange de la vraisemblance, dont chaque composante décrit un régime de fonctionnement du capteur, ce qui fait le lien avec notre approche. Cependant, la manière de faire est totalement différente, et consiste à estimer l'indice du régime via le filtre particulaire, alors que notre modèle n'ajoute pas de dimension à l'espace d'état, et se sert uniquement de celui-ci pour définir l'état de fonctionnement d'un capteur, comme nous le verrons dans la section 6.2.2. De plus, le champ d'application de notre approche semble plus vaste, puisqu'elle peut être appliquée à des problématiques mono-modales, multi-capteurs, ou multi-caractéristiques (ces extensions ne sont pas discutées dans [Caron et al., 2007]). L'esprit de la méthode diffère également, car nous préférons parler de vraisemblances à modèles de composantes multiples, dont les apparences sont des cas particuliers, que d'états de fonctionnement d'un capteur.

6.2.2 Modèle proposé

Décrivons maintenant la formalisation de l'approche proposée. Soit $\mathbf{\hat{x}} = (\mathbf{\hat{x}}^1, \dots, \mathbf{\hat{x}}^O)$ un vecteur de O composantes, où $\mathbf{\hat{x}}^j$ représente le modèle de référence, ou l'apparence, de l'objet associé à la *j*-ème composante, elle-même désignant une posture. Par exemple, si *j* = 2 représente la posture *derrière* du visage (composante définie grâce à une information sur l'orientation du visage), un modèle d'apparence de couleurs $\mathbf{\hat{x}}^2$ sera majoritairement décrit par la couleur des cheveux. La vraisemblance jointe s'écrit :

$$p(\mathbf{y}_{t}|\mathbf{x}_{t}, \mathbf{\hat{x}}) = \sum_{j=1}^{O} p(\mathbf{y}_{t}, \mathbf{l} = j | \mathbf{x}_{t}, \mathbf{\hat{x}}^{j})$$
$$= \sum_{j=1}^{O} p(\mathbf{l} = j | \mathbf{x}_{t}, \mathbf{y}_{t}) p(\mathbf{y}_{t} | \mathbf{x}_{t}, \mathbf{\hat{x}}^{j})$$
$$= \sum_{j=1}^{O} \varphi^{j} (\gamma(\mathbf{x}_{t}, \mathbf{y}_{t})) p(\mathbf{y}_{t} | \mathbf{x}_{t}, \mathbf{\hat{x}}^{j})$$
(6.8)

avec $p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{x}^j)$ la vraisemblance de la *j*-ème décomposition et $\varphi^j(\gamma(\mathbf{x}_t, \mathbf{y}_t))$ le poids de la *j*-ème posture associée à la caractéristique $\gamma(\mathbf{x}_t, \mathbf{y}_t)$, avec

$$\varphi^{j}: \mathcal{Z} \to [0,1] \quad \text{tel que} \quad \forall (\mathbf{x}_{t}, \mathbf{y}_{t}) \in \mathcal{X} \times \mathcal{Y}, \sum_{i=1}^{O} \varphi^{i} \left(\gamma(\mathbf{x}_{t}, \mathbf{y}_{t}) \right) = 1$$
 (6.9)

avec \mathcal{Z} l'espace des caractéristiques, et $\gamma : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ la fonction des caractéristiques. On appelle $\mathbf{l} = j$ la *j*-ème posture de la décomposition, et $\varphi^j(\gamma(\mathbf{x}_t, \mathbf{y}_t))$ la probabilité de la posture *j*. Afin de proposer un modèle plus général, nous introduisons une nouvelle notation, en différenciant l'observation utilisée dans $\varphi^j(\gamma(\mathbf{x}_t, \mathbf{y}_t))$ de celle de la vraisemblance $p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{\hat{x}}^j)$. Le vecteur d'observation \mathbf{y}_t possède donc deux composantes, $\mathbf{y}_t = (\mathbf{y}_t^{Pos}, \mathbf{y}_t^{App})$, où \mathbf{y}_t^{Pos} est l'observation servant à définir les probabilités des postures du modèle, et \mathbf{y}_t^{App} l'observation utilisée dans la vraisemblance. En reprenant l'exemple du suivi de visage en 3D, les probabilités des postures, utilisant un critère d'orientation de la tête, pourraient être calculées grâce à des informations extraites de l'image du gradient, alors que les vraisemblances conditionnées aux apparences pourraient être définies par une notion de distance entre histogrammes de couleurs. En pratique, la distinction entre observations associées aux postures et aux apparences n'est pas toujours nécessaire, puisque si $\mathbf{y}_t^{Pos} = \mathbf{y}_t^{App}$, nous ne modélisons en réalité qu'une seule observation, et le modèle revient à celui présenté dans l'équation 6.8. Les probabilités des postures peuvent ne pas dépendre d'une observation, ainsi $\varphi^j(\gamma(\mathbf{x}_t, \mathbf{y}_t)) = \varphi^j(\gamma(\mathbf{x}_t))$, cas que nous étudierons également dans la section 6.4. Cependant, dans un cadre général, nous faisons cette distinction, la vraisemblance à modèles multiples s'écrit alors :

$$p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{\hat{x}}) \simeq \sum_{j=1}^{O} \varphi^j \left(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos}) \right) p(\mathbf{y}_t^{App} | \mathbf{x}_t, \mathbf{\hat{x}}^j)$$
(6.10)

Cette équation est obtenue en considérant l'indépendance conditionnelle de l par rapport à \mathbf{y}_t^{App} sachant \mathbf{y}_t^{Pos} , et le terme $p(\mathbf{y}_t^{Pos}|\mathbf{x}_t, \mathbf{x}^j, \mathbf{y}_t^{App})$ est simplement ignoré, puisque nous ne savons pas, et ne souhaitons pas, modéliser cette vraisemblance. Ainsi, cette approximation sert à mettre en évidence la décomposition « posture » et « apparence » étudiée, et ne requiert pas l'existence de \mathbf{y}_t^{Pos} , ce qui nous permet de facilement se ramener à l'écriture proposée dans l'équation 6.8.

Le caractère original de la décomposition effectuée dans l'équation 6.10 est le conditionnement de la probabilité de l par rapport à \mathbf{x}_t et \mathbf{y}_t^{Pos} . Cela signifie que les poids sont automatiquement déterminés par la dernière observation, et pour la particule considérée, ce qui s'oppose aux méthodes de la littérature présentées dans la section 6.1. La procédure de filtrage particulaire pour l'intégration de modèles d'apparences multiples est présentée dans l'algorithme 6.1.

Le rapprochement entre la contribution proposée dans ce chapitre et le fil conducteur de ce manuscrit ressemble d'ailleurs à celui existant entre le modèle proposé dans [Caron et al., 2007] et l'approche développée ici. Nous avons proposé, dans le chapitre 3, d'intégrer, par un modèle à sauts, des informations spatiales floues dans le filtre particulaire. Chaque information spatiale décrit une partition de l'espace des paramètres, dont chaque élément est un ensemble flou. Comme nous l'avons vu dans la section 4.1.3, à propos de la modélisation de la vraisemblance de la bouche, l'ensemble flou désignant la posture conditionne l'expression de la vraisemblance, en définissant un modèle d'histogramme par posture floue. L'approche est bien sûr plus générale que cela, la vraisemblance pouvant être conditionnée par n'importe quel type d'information spatiale. Mais le point intéressant est que le modèle à sauts de posture peut également s'interpréter par une densité de mélange de la vraisemblance, chaque poids correspondant alors à la probabilité définie de considérer ladite posture. En liant une apparence à une posture de l'objet, le lien devient donc évident, même si, une fois encore, la mise en œuvre diffère totalement, l'approche présentée dans ce chapitre pouvant s'adapter à n'importe quel type de modèle. L'avantage de ne pas considérer la posture directement dans l'espace d'état permet de ne pas ajouter une dimension à celui-ci, ce qui, comme nous en avons notamment déjà **Algorithme 6.1:** Filtre particulaire pour l'intégration de modèles d'apparences multiples.

Entrée : Approximation initiale $\mathbb{P}(d\mathbf{x}_0) \simeq \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{x}_0^{(n)}}(d\mathbf{x}_0)$ **Sortie** : Approximation P_N de la loi *a posteriori* de la trajectoire $\mathbb{P}(d\mathbf{x}_{0:T}|\mathbf{y}_{1:T})$ début pour $t = 1, \ldots, T$ faire • pour $n = 1, \ldots, N$ faire • Générer $\tilde{\mathbf{x}}_{t}^{(n)} \sim q(\mathbf{x}_{t}|\mathbf{x}_{0:t-1}^{(n)}, y_{t})$ • Poser $\tilde{\mathbf{x}}_{0:t}^{(n)} = (\mathbf{x}_{0:t-1}^{(n)}, \tilde{\mathbf{x}}_{t}^{(n)})$ • Calculer les poids d'importance : $\tilde{w}(\tilde{\mathbf{x}}_{t}^{(n)}) = w_{t-1}^{(n)} \frac{\left[\sum_{j=1}^{O} \varphi^{j} \left(\gamma(\tilde{\mathbf{x}}_{t}^{(n)}, \mathbf{y}_{t}^{Pos})\right) p(\mathbf{y}_{t}^{App} | \tilde{\mathbf{x}}_{t}^{(n)}, \tilde{\mathbf{x}}_{j}^{j})\right] p(\tilde{\mathbf{x}}_{t}^{(n)} | \mathbf{x}_{t-1}^{(n)})}{q(\tilde{\mathbf{x}}_{t}^{(n)} | \mathbf{x}_{0:t-1}^{(n)}, \mathbf{y}_{1:t})}$ • pour $n = 1, \ldots, N$ faire • Calculer les poids d'importance normalisés : $w_t^{*(n)} = \frac{\tilde{w}(\tilde{\mathbf{x}}_t^{(n)})}{\sum_{m=1}^N \tilde{w}(\tilde{\mathbf{x}}_t^{(m)})}$ • si $\hat{N}_{eff}(N) < N_{seuil}(N)$ alors • Rééchantillonner $\{\mathbf{x}_{0:t}^{(n)}, 1/N\}_{n=1}^N$ à partir du nuage $\{\tilde{\mathbf{x}}_{0:t}^{(n)}, w_t^{*(n)}\}_{n=1}^N$ selon l'algorithme du rééchantillonnage multinomial 1.9. sinon $\circ \ \operatorname{Poser} \ \{\mathbf{x}_{0:t}^{(n)}, w_t^{(n)}\}_{n=1}^N = \{\tilde{\mathbf{x}}_{0:t}^{(n)}, w_t^{*(n)}\}_{n=1}^N$ • Approximation de la loi marginale au temps *t*, dite loi de filtrage : $\mathbb{P}(d\mathbf{x}_t|\mathbf{y}_{1:t}) \simeq P_N(d\mathbf{x}_t|\mathbf{y}_{1:t}) = \sum_{t=1}^N w_t^{(n)} \delta_{\mathbf{x}_t^{(n)}}(d\mathbf{x}_t)$ • Espérance de Monte-Carlo issue de la loi de filtrage : $\mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[\phi(\mathbf{x}_t)] \simeq \sum_{n=1}^N w_t^{(n)} \phi(\mathbf{x}_t^{(n)})$ retourner $P_N(d\mathbf{x}_{0:T}|\mathbf{y}_{1:T}) = \sum_{n=1}^N w_T^{(n)} \delta_{\mathbf{x}_{0:T}^{(n)}}(d\mathbf{x}_{0:T})$

173

discuté dans la section 1.3.1.4, augmente la variance des poids des particules.

6.2.3 Applications

Comme nous l'avons vu, les modèles d'apparence \mathbf{x} conditionnent la vraisemblance, et sont supposés connus et fixés. Cependant, l'approche proposée peut également être adaptée au cas où la vraisemblance n'utilise pas de modèles d'apparence. Par exemple, en décomposant la vélocité d'un objet par la fonction $\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos})$ en un nombre fini de postures, disons neuf, chaque poids $\varphi^j(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos}))$ désignant la probabilité que l'objet soit dans une des neufs vélocités, la vraisemblance $p^j(\mathbf{y}_t^{App}|\mathbf{x}_t)$ de déplacement pourrait utiliser la projection de la *j*-ème vélocité sur l'image vectorielle du flot optique. Le modèle correspondant s'écrit alors :

$$p(\mathbf{y}_t|\mathbf{x}_t) = \sum_{j=1}^{O} \varphi^j \left(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos}) \right) p^j(\mathbf{y}_t^{App}|\mathbf{x}_t)$$
(6.11)

Cette décomposition est plus générale, car elle ne fait pas intervenir de notion d'apparence. Cependant, dans le cadre de ce travail, nous considérons un modèle multiple d'apparences, et donc la décomposition définie dans l'équation 6.10.

Bien que le champ d'application semble vaste, nous choisissons, comme étude préliminaire de ce modèle, de définir des poids $\varphi^j \left(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos}) \right)$ qui dépendent à la fois de l'état \mathbf{x}_t et de l'observation \mathbf{y}_t , dans le but de montrer le potentiel de modélisation de l'approche. De plus, bien que ces poids puissent être, dans nombre d'applications, définis par une fonction analytique γ connue, nous nous intéressons au cas où ils sont définis grâce à un algorithme d'apprentissage. Cet algorithme, dans le cas où la base d'apprentissage n'est pas étiquetée, peut être une technique de *clustering* de type k-moyennes [Lloyd, 1982; Dhillon et al., 2004], ou de type Support Vector Clustering [Ben-Hur et al., 2002]. Cependant, dans le cadre de ce travail, les étiquettes de la base d'apprentissage, qui correspondent en fait aux indices désignant les postures, sont supposées connues. La méthodologie consiste donc à utiliser un classifieur supervisé, qui dans notre cas est une méthode de machine à vecteurs de support multi-classes et probabiliste (SVM), dans l'espace des caractéristiques \mathcal{Z} , ce qui permet de calculer la probabilité d'appartenance d'un couple $(\mathbf{x}_t, \mathbf{y}_t^{Pos})$ à une classe j grâce à la fonction de caractéristiques γ . Le choix d'un SVM s'explique par son efficacité, sa généricité, et sa faculté à pouvoir extraire une sortie de classification probabiliste, qui sera ensuite intégrée dans notre modèle.

L'objet de la prochaine section est la construction d'un classifieur SVM, selon la caractéristique utilisée, et selon les classes considérées. Cela nous permettra, pendant la phase de suivi, de calculer la probabilité d'appartenance à une classe $j \varphi^j (\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos}))$ d'une valeur de caractéristique $\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos})$ associée au couple $(\mathbf{x}_t, \mathbf{y}_t^{Pos})$.

6.3 Machine à vecteurs de support (SVM)

L'objectif d'une machine à vecteurs de support (SVM en anglais, pour *Support Vector Machine*), également appelée séparateur à vaste marge, est de construire un hyperplan, dans un espace à déterminer, qui sépare un ensemble de données en deux classes. La stratégie consiste à maximiser la distance de séparation entre le plan et les données les plus proches. Cette distance s'appelle la marge. L'hyper-espace utilisé pour la classification est déterminé par la fonction de projection des données d'entrée, que l'on choisit

implicitement en définissant une fonction semi-définie positive dite « fonction noyau » calculée pour toutes les paires de la base d'apprentissage.

Dans le cadre du suivi par filtrage particulaire, l'utilisation d'un SVM reste marginale, car souvent très spécifique. Dans [Ye et al., 2009], les auteurs proposent de l'utiliser comme détecteur dans la fonction de proposition dans un contexte de suivi de véhicule. Dans [Patras et Hancock, 2010], les auteurs utilisent le classifieur pour déterminer la fiabilité d'une nouvelle observation, et intégrer celle-ci dans le filtre particulaire. Dans [Liangcheng et Huizhong, 2009], le filtre particulaire est utilisé pour estimer le jeu de paramètres, qualifié par les auteurs d'idéal, du SVM.

Dans notre approche, nous souhaitons utiliser un SVM pour déterminer automatiquement les poids d'appartenance aux classes, c'est-à-dire aux postures. L'objectif est donc de définir un SVM multi-classes, avec une sortie probabiliste.

6.3.1 Principe du SVM à deux classes

Nous supposons l'entrée d'un problème d'apprentissage de la forme $\mathcal{D} = \{(z^{(1)}, l^{(1)}), \ldots, (z^{(M)}, l^{(M)})\}$, de paires $(z^{(i)}, l^{(i)}) \in \mathcal{Z} \times \{-1, 1\}$, avec $z^{(i)}$ la donnée (l'exemple, ou l'observation) et $l^{(i)}$ son étiquette (binaire), qui sont utilisées pour apprendre une fonction $f_{\mathcal{D}} : \mathcal{Z} \to \{-1, 1\}$, qui attribue une étiquette à une donnée. Dans le cadre d'un apprentissage statistique, nous faisons l'hypothèse que les paires contenues dans la base d'apprentissage \mathcal{D} , tout comme les paires de test (z, l), sont décrites par la même probabilité inconnue \mathbb{P} sur $\mathcal{Z} \times \{-1, 1\}$.

L'objectif est de trouver une fonction qui minimise l'estimation du risque empirique, dans le but d'effectuer une classification binaire. La forme à marge souple proposée par [Cortes et Vapnik, 1995] tolère les mauvaises classifications, en introduisant les variables ressorts (*slack variables* en anglais) $\xi = (\xi_1, \ldots, \xi_M)$ qui relâchent les contraintes initiales de la manière suivante : $l^{(i)} (\langle w, \Phi(z^{(i)}) \rangle + b) \ge 1 - \xi_i$, avec $\Phi : \mathbb{Z} \to \mathcal{F}$ une fonction non linéaire, et \mathcal{F} l'espace des caractéristiques du SVM, aussi appelé espace de redescription. Ajouter les variables ressorts dans la fonction objectif permet de garder leurs valeurs relativement basses, afin de pénaliser les changements par rapport aux contraintes initiales. Le problème d'optimisation correspondant s'écrit :

$$\begin{cases}
\min_{\substack{w \in \mathcal{F}, b \in \mathbb{R}, \xi \in \mathbb{R}^{M}}} \frac{1}{2} \langle w, w \rangle + C \sum_{i=1}^{M} \xi_{i} \text{ sous les contraintes} \\
l^{(i)} \left(\langle w, \Phi(z^{(i)}) \rangle + b \right) \ge 1 - \xi_{i}, \quad \xi_{i} \ge 0, \quad i = 1, \dots, M
\end{cases}$$
(6.12)

où C > 0 est un paramètre fixé qui pondère le second terme de la fonction objectif par rapport au premier. Plus C est faible, moins le classifieur pénalise les données aberrantes qu'il va choisir d'ignorer dans le but d'optimiser la marge.

Le problème d'optimisation 6.12 se situe dans l'espace des caractéristiques \mathcal{F} du SVM, qui est souvent multi-dimensionnel, ou de dimension infinie. Il est possible de le résoudre en utilisant des multiplicateurs de Lagrange $\alpha = (\alpha_1, \ldots, \alpha_M) \ge 0$ pour les contraintes $l^{(i)}(\langle w, \Phi(z^{(i)}) \rangle + b) \ge 1 - \xi_i$ et $\beta = (\beta_1, \ldots, \beta_M) \ge 0$ pour les contraintes $\xi_i \ge 0$:

$$\mathcal{L}(w,b,\xi,\alpha,\beta) = \min_{w,b,\xi} \max_{\alpha,\beta} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^M \xi_i - \sum_{i=1}^M \alpha_i \left[\xi_i - 1 + l^{(i)} (w^T \Phi(z^{(i)}) + b)\right] - \sum_{i=1}^M \beta_i \xi_i$$
(6.13)

Dans le but de trouver l'optimimum de l'équation 6.12, nous avons besoin de minimiser le lagrangien par rapport à (w, b, ξ) :

$$\frac{\partial \mathcal{L}}{\partial w}(w, b, \xi, \alpha, \beta) = w - \sum_{i=1}^{M} l^{(i)} \alpha_i \Phi(z^{(i)}) = 0$$
(6.14)

$$\frac{\partial \mathcal{L}}{\partial b}(w, b, \xi, \alpha, \beta) = \sum_{i=1}^{M} l^{(i)} \alpha_i = 0$$
(6.15)

$$\frac{\partial \mathcal{L}}{\partial \xi_i}(w, b, \xi, \alpha, \beta) = C - \alpha_i - \beta_i = 0, \quad i = 1, \dots, M.$$
(6.16)

En injectant les équations 6.14 et 6.15 dans l'équation 6.13, nous obtenons la formulation du problème dual :

$$\begin{cases} \max_{\alpha} \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_i \alpha_j l^{(i)} l^{(j)} K(z^{(i)}, z^{(j)}) & \text{sous les contraintes} \\ \alpha \in [0, C]^M \\ \sum_{i=1}^{M} l^{(i)} \alpha_i = 0 \end{cases}$$

$$(6.17)$$

où la contrainte $\alpha \in [0, C]^M$ est une implication de l'équation 6.16, et où $K(z^{(i)}, z^{(j)}) = \Phi(z^{(i)})^T \cdot \Phi(z^{(j)})$ est la fonction noyau, qui doit être connue, et qui permet de considérer des espaces de caractéristiques complexes du SVM, sans avoir à définir la fonction Φ . Le choix du noyau est de première importance, puisqu'en choisissant un noyau linéaire, c'est-à-dire $K(z^{(i)}, z^{(j)}) = (z^{(i)})^T \cdot (z^{(j)})$, la minimisation est effectuée dans l'espace de départ, et le classifieur ainsi construit est linéaire. A l'inverse, avec un noyau gaussien $K(z^{(i)}, z^{(j)}) = \exp(-||z^{(i)} - z^{(j)}||^2 / [2\sigma^2])$, l'espace des caractéristiques \mathcal{F} du SVM devient hilbertien de dimension infinie, ce qui permet d'opérer la séparation linéaire directement dans ce nouvel espace (et la séparation dans l'espace de départ peut être plus complexe).

La quantité α peut être calculée en utilisant l'une des nombreuses méthodes proposées dans la littérature [Keerthi et al., 2001; Fan et al., 2005; Chen et al., 2006]. Ensuite, β est calculé en utilisant $\beta_i = C - \alpha_i, \forall i = 1, ..., M$. Pour calculer *b*, nous utilisons les conditions de Karush-Kuhn-Tucker pour les multiplicateurs de Lagrange. Ainsi, pour un certain *i* avec $0 < \alpha_i < C$, les contraintes $\xi_i = 0$ et $\xi_i - 1 + l^{(i)}(w^T \Phi(z^{(i)}) + b) = 0$ sont vérifiées, ce qui nous donne :

$$b = l^{(i)} - w^T \Phi(z^{(i)}) = l^{(i)} - \sum_{j=1}^M l^{(j)} \alpha_j K(z^{(j)}, z^{(i)})$$
(6.18)

Dans le but de prédire la classe d'un nouvel échantillon $z \in Z$, il suffit de calculer la réponse du classifieur :

$$f_{\mathcal{D}}^{*}(z) = w^{T} \Phi(z) + b = \sum_{j=1}^{M} l^{(j)} \alpha_{j} K(z^{(i)}, z) + b$$
(6.19)

puis de déduire son appartenance selon le signe de $f_D(z)$, c'est-à-dire $f_D(z) = 1$ si $f_D^*(z) > 0$, -1 sinon.

6.3.2 Extensions à plusieurs classes

Les extensions majeures classiques de la classification binaire fondamentale traitent de la classification multi-classes et/ou avec des sorties probabilistes. L'approche typique est d'utiliser un ensemble de classifieurs binaires dans le but de résoudre un problème multi-classes donné, avec L le nombre de classes. Dans les paragraphes suivants, nous passons brièvement en revue les stratégies les plus répandues.

6.3.2.1 Un contre tous (WTA)

La stratégie WTA [Schölkopf et al., 1995] (pour *Winner-Takes-All* en anglais) utilise *L* classifieurs, le *i*-ème d'entre eux étant entraîné avec des exemples de la *i*-ème classe contre tous les autres éléments. Un nouvel élément *z* est assigné à la classe dont la marge est la plus élevée, c'est-à-dire à $\arg \max f_i^*(z)$.

6.3.2.2 Vainqueur par maximum de victoires (MWS)

Avec la stratégie MWS [Knerr et al., 1990] (pour *Max-Wins Voting* en anglais), C_L^2 classifieurs binaires sont utilisés pour chaque paire de classes, et un vote est effectué par chaque classifieur. La stratégie sous-jacente consiste à assigner à un nouvel élément z la classe qui recevra le plus grand nombre de votes de la part des C_L^2 classifieurs binaires.

6.3.2.3 Stratégie Pairwise Coupling

Si les sorties SVM sont interprétées comme des probabilités *a posteriori* [Platt, 2000], nous notons r_{ij} les probabilités estimées entre les classes *i* et *j* telles que $r_{ij} \simeq \mathbb{P}(l = i|l = i \text{ ou } j, z)$. En utilisant cette information, le but est de calculer les probabilités $p_i = \mathbb{P}(l = i|z), i = 1, ..., L$, telles que $r_{ij} = p_i/(p_i + p_j)$. En considérant l'égalité $\mathbb{P}(l = i|l = i \text{ ou } j, z) \mathbb{P}(l = j|z) = \mathbb{P}(l = j|l = i \text{ ou } j, z) \mathbb{P}(l = i|z)$, soit $r_{ij}p_j = r_{ji}p_i$, nous obtenons un système de *L* inconnues avec \mathbf{C}_L^2 équations. Différentes solutions ont été proposées dans le but d'obtenir les probabilités p_i qui nous intéressent, et nous adoptons celle classiquement utilisée dans la littérature et décrite dans [Wu et al., 2004], qui optimise par différences quadratiques le problème convexe suivant :

$$\begin{cases} \min_{p} \sum_{i=1}^{L} \sum_{j \neq i} (r_{ij}p_j - r_{ji}p_i)^2 \\ \sum_{i=1}^{L} p_i = 1, p_i \ge 0 \end{cases}$$
(6.20)

6.3.3 Intégration dans une vraisemblance à modèles multiples

Reprenons la vraisemblance à modèles multiples définie dans la section 6.2 :

$$p(\mathbf{y}_t | \mathbf{x}_t, \mathbf{\hat{x}}) = \sum_{j=1}^{O} \varphi^j \left(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos}) \right) \, p(\mathbf{y}_t^{App} | \mathbf{x}_t, \mathbf{\hat{x}}^j)$$

Nous souhaitons déterminer les poids des postures $\varphi^j \left(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos})\right)$ par un SVM. En utilisant une stratégie *Pairwise Coupling*, nous sommes capables de définir des probabilités d'appartenance $p_j = \mathbb{P}(l = j|z), j = 1, ..., L$. Dans le modèle proposé dans l'équation 6.10, la valeur p_j correspond donc à la probabilité de considérer la *j*-ème posture du modèle, c'est-à-dire à $\varphi^j \left(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos})\right)$ (la *j*-ème posture du modèle étant liée à la *j*-ème apparence). Dans un problème avec une décomposition en O éléments, le but est alors de calculer les probabilités d'appartenance liées à chacune des composantes, $p_j = \mathbb{P}(l = j|z), j = 1, \ldots, O$. Il suffit donc d'entraîner un SVM avec des éléments d'apprentissage $\mathcal{D} = \left\{ \left(\gamma(\mathbf{x}^{(1)}, \mathbf{y}^{Pos,(1)}), l^{(1)}\right), \ldots, \left(\gamma(\mathbf{x}^{(M)}, \mathbf{y}^{Pos,(M)}), l^{(M)}\right) \right\}$, avec γ la fonction des caractéristiques (qui correspond par exemple à un histogramme de couleurs, ou à un ensemble de points de contrôle qui définit une forme), et $l^{(M)}$ l'étiquette, qui indique l'indice de la posture considérée, et désigne donc implicitement l'apparence utilisée. Les poids $\varphi^j \left(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos})\right)$ sont ensuite simplement obtenus en calculant les probabilités $\{p_j\}_{j=1}^O$ telles que $\forall j \in \{1, \ldots, O\}, \varphi^j \left(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos})\right) = \mathbb{P}(\mathbf{l} = j | \gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos})) = p_j$. Nous pouvons également utiliser les probabilités d'appartenance des échantillons de

Nous pouvons également utiliser les probabilités d'appartenance des échantillons de la base de données aux classes afin de construire les modèles d'appartenance $\{\mathbf{x}^{j}\}_{j=1}^{O}$. Ils sont définis comme la somme de tous les échantillons pondérée par leur probabilité d'appartenance à la composante considérée :

$$\mathbf{\hat{x}}^{j} = \frac{\sum_{i=1}^{M} \mathbb{P}\left(\mathbf{l} = j | \gamma(\mathbf{x}^{(i)}, \mathbf{y}^{Pos,(i)})\right) \eta(\mathbf{x}^{(i)})}{\sum_{i=1}^{M} \mathbb{P}\left(\mathbf{l} = j | \gamma(\mathbf{x}^{(i)}, \mathbf{y}^{Pos,(i)})\right)}$$
(6.21)

où $\mathbb{P}\left(\mathbf{l} = j | \gamma(\mathbf{x}^{(i)}, \mathbf{y}^{Pos,(i)})\right)$ est la probabilité de considérer la posture j selon la donnée $\gamma(\mathbf{x}^{(i)}, \mathbf{y}^{Pos,(i)})$, que l'on extrait d'une stratégie de *Pairwise Coupling* (section 6.3.2.3), et η est la fonction caractérisant une apparence, qui sera explicitée dans la section 6.4. La procédure du filtre particulaire avec l'utilisation d'un SVM est décrite dans l'algorithme 6.2.

6.4 Expériences

Comme première application, nous choisissons celle du suivi de bouche, déjà étudiée dans le chapitre 4, en utilisant une, deux, et trois postures (décomposition *bouche*, ou (*bouche fermée*, *bouche ouverte*), ou encore (*bouche fermée*, *bouche ouverte*, *sourire*)). Les densités de probabilités $p^{j}(\mathbf{y}_{t}|\mathbf{x}_{t})$ sont des vraisemblances de couleurs, via l'apprentissage d'histogrammes de référence. Les expériences que nous présenterons dans la section 6.4.2 utiliseront successivement des poids de postures définies par des caractéristiques d'histogramme de couleurs avec une distance de Bhattacharyya, de forme avec une distance euclidienne, et de forme avec une distance de Hausdorff [Hausdorff, 1962].

6.4.1 Protocole de tests

6.4.1.1 Présentation de la séquence

La séquence de suivi de bouche utilisée dans ce chapitre a été présentée dans la section 4.1.3. Elle contient 5000 images, montrant le visage d'un individu dont les expressions faciales changent (figure 6.2). La base de données pour l'apprentissage SVM comporte 1000 éléments. Chaque élément est une forme de bouche donnée par un ensemble de *P* points de contrôle, dont on extrait l'histogramme de couleurs. Nous notons $\mathcal{D} = \{(\gamma(z^{(1)}), l^{(1)}), \dots, (\gamma(z^{(M)}), l^{(M)})\}\$ les *M* échantillons de la base de données, avec $\gamma(z^{(i)})$ la fonction de caractéristique utilisée dans le SVM (dont l'écriture sera explicitée dans la section 6.4.2) extrait de la *i*-ème forme $z^{(i)}$.

Algorithme 6.2: Filtre particulaire pour l'intégration de modèles d'apparences multiples, avec l'utilisation d'un SVM pour calculer les poids des composantes de la densité de mélange de la vraisemblance.

Entrée : Approximation initiale $\mathbb{P}(d\mathbf{x}_0) \simeq \frac{1}{N} \sum_{n=1}^{N} \delta_{\mathbf{x}_0^{(n)}}(d\mathbf{x}_0)$ Base d'apprentissage $\mathcal{D} = \left\{ \left(\gamma(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), l^{(1)} \right), \dots, \left(\gamma(\mathbf{x}^{(M)}, \mathbf{y}^{(M)}), l^{(M)} \right) \right\}$ Sortie : Approximation P_N de la loi *a posteriori* de la trajectoire $\mathbb{P}(d\mathbf{x}_{0:T} | \mathbf{y}_{1:T})$ début 1. Apprendre les \mathbf{C}_O^2 classifieurs SVM grâce aux échantillons de \mathcal{D} . 2. Calculer les probabilités $\left\{ \mathbb{P} \left(\mathbf{l} = j | \gamma(\mathbf{x}^{(i)}, \mathbf{y}^{Pos,(i)}) \right) \right\}_{j=1..O, i=1..M}$ en utilisant une stratégie *Pairwise Coupling*. 3. Calculer les modèles d'apparence $\{\mathbf{x}^j\}_{j=1}^O$ selon l'équation 6.21. 4. pour $t = 1, \dots, T$ faire • pour $n = 1, \dots, N$ faire • Pour un échantillon généré $\tilde{\mathbf{x}}_t^{(n)}$, et une observation \mathbf{y}_t^{Pos} , calculer les probabilités $\varphi^j \left(\gamma(\tilde{\mathbf{x}}_t^{(n)}, \mathbf{y}_t^{Pos}) \right) = \mathbb{P} \left(\mathbf{l} = j | \gamma(\tilde{\mathbf{x}}_t^{(n)}, \mathbf{y}_t^{Pos}) \right)$ et les intégrer dans la vraisemblance du filtre particulaire (algorithme 6.1). retourner $P_N(d\mathbf{x}_{0:T} | \mathbf{y}_{1:T}) = \sum_{n=1}^N w_T^{(n)} \delta_{\mathbf{x}_0^{(n)}}(d\mathbf{x}_{0:T})$

Nous nous proposons de suivre la bouche dans une séquence de 300 images (hors base d'apprentissage). Comme base de comparaison, nous considerons trois décompositions de vraisemblance de la bouche : une décomposition d'un élément, *bouche*, ce qui revient au cas classique (aucune classification) ; une décomposition en deux éléments, *bouche fermée* et *bouche ouverte* ; et enfin une décomposition en trois éléments, *bouche fermée*, *bouche ouverte*, et *sourire*.



FIG. 6.2 – Images de la séquence de suivi de bouche, montrant des exemples des trois classes de bouche (a) *fermée*, (b) *ouverte* et (c) *sourire*.

6.4.1.2 Modèle dynamique

Le vecteur d'état $\mathbf{x}_t = (\mathbf{x}_t, \mathbf{y}_t, \dot{\mathbf{x}}_t, \dot{\mathbf{y}}_t, \theta_t, \mathbf{a}_t)$ contient les coordonnées 2D $(\mathbf{x}_t, \mathbf{y}_t)$ du centre de la bouche au temps t, le vecteur de déplacement $(\dot{\mathbf{x}}_t, \dot{\mathbf{y}}_t)$, l'orientation de l'objet θ_t et un vecteur de points de contrôle $\mathbf{a}_t = (\mathbf{a}_t^1, \dots, \mathbf{a}_t^P)$. Nous considérons le modèle

dynamique présenté dans l'équation 4.1.

Le modèle dynamique utilisé est le même que le modèle flou proposé dans la section 4.1.3, pour lequel nous avons considéré neuf ensembles flous pour le paramètre de vélocité, cinq pour l'orientation de l'objet et trois pour les postures floues. Nous renvoyons le lecteur à la section correspondante pour plus de détails.

Notons que l'utilisation d'un modèle flou n'est évidemment pas obligatoire, l'approche proposée dans ce chapitre convient à tous types de dynamiques et d'états.

6.4.1.3 Modèle de vraisemblance

Comme dans la section 4.1.3, nous utiliserons une vraisemblance $p(\mathbf{y}_t^{App}|\mathbf{x}_t, \mathbf{\hat{x}}^j)$ combinant une information de couleurs et de contours :

$$p(\mathbf{y}_t^{App} | \mathbf{x}_t, \mathbf{\hat{x}}^j) = p(\mathbf{y}_t^{App, R} | \mathbf{x}_t, \mathbf{\hat{x}}^j) \, p(\mathbf{y}_t^{App, C} | \mathbf{x}_t)$$

où $p(\mathbf{y}_t^{App,C}|\mathbf{x}_t)$ correspond à la vraisemblance de contour définie dans l'équation 4.10, et $p(\mathbf{y}_t^{App,R}|\mathbf{x}_t, \hat{\mathbf{x}}^j)$ la vraisemblance de région, conditionnée au *j*-ème modèle de référence. La vraisemblance de région utilise la distance entre histogrammes de couleurs HSV (pour plus d'informations, voir la section 4.1.3) [Pérez et al., 2002]. Nous notons a_t l'ensemble des pixels appartenant à l'objet, décrit par le contour a_t , à un instant *t*. La première distance mesure la similarité entre le *j*-ème histogramme de référence $\hat{\mathbf{x}}^j$ et un histogramme candidat défini par l'état \mathbf{x} et les informations de couleur de l'image \mathbf{y}_t^{App} , que nous notons $h(a_t) = \{h(a_t; u)\}_{u=1}^U$, avec *u* l'indice d'une case d'un histogramme de longueur *U*. La seconde composante du calcul est la distance entre l'histogramme candidat et l'histogramme $h_{BG}^n = \{h_{BG}^n(u)\}_{u=1}^U$ de l'arrière plan de l'image :

$$p(\mathbf{y}_t^{App,R}|\mathbf{x}_t, \mathbf{\hat{x}}^j) \propto \exp\left(-\lambda_2(d^2[\mathbf{\hat{x}}^j, h(\mathring{a}_t)] - d^2[h_{BG}^{\bullet}, h(\mathring{a}_t)])\right)$$
(6.22)

avec d^2 la distance de Bhattacharyya définie dans l'équation 4.12. L'histogramme du fond de l'image est supposé connu (il est appris sur la première image de la séquence). Les histogrammes de référence $\{\mathbf{x}^j\}_{j=1}^O$ sont calculés automatiquement grâce au SVM (la construction de celui-ci sera expliquée dans la section 6.4.2), tel que cela a été proposé dans l'équation 6.21. L'histogramme de référence de la *j*-ème apparence est donc calculé selon :

$$\mathbf{x}^{j} = \frac{\sum_{i=1}^{M} \mathbb{P}\left(\mathbf{l} = j | \gamma(z^{(i)})\right) h(z^{(i)})}{\sum_{i=1}^{M} \mathbb{P}\left(\mathbf{l} = j | \gamma(z^{(i)})\right)}$$

$$(6.23)$$

où $\mathbb{P}\left(\mathbf{l} = j | \gamma(z^{(i)})\right)$ est la probabilité de considérer la classe j selon la donnée $\gamma(z^{(i)})$, que l'on extrait d'une stratégie de *Pairwise Coupling* (section 6.3.2.3), $\gamma(z^{(i)})$ la fonction de caractéristiques, qui sera explicitée dans la section 6.4.2, et $h(z^{(i)})$ l'histogramme extrait de l'échantillon $z^{(i)}$ de la base de données.

La vraisemblance jointe en considérant ${\cal O}$ composantes pour la vraisemblance de couleurs s'écrit finalement :

$$p(\mathbf{y}_t|\mathbf{x}_t, \mathbf{\hat{x}}) = p(\mathbf{y}_t^{App,C}|\mathbf{x}_t) \sum_{j=1}^{O} \varphi^j \left(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos})\right) p(\mathbf{y}_t^{App,R}|\mathbf{x}_t, \mathbf{\hat{x}}^j)$$
(6.24)

Il ne nous reste plus qu'à définir la fonction γ , afin de construire notre classifieur.
6.4.2 Apprentissage SVM

Concernant l'apprentissage du SVM, nous avons considéré des noyaux gaussiens, afin d'être en mesure d'obtenir une séparation non linéaire de nos données :

$$K(\gamma(z^{(i)}), \gamma(z)) = \exp\left(-\frac{\langle \gamma(z^{(i)}), \gamma(z) \rangle^2}{2\sigma^2}\right)$$
(6.25)

avec $z^{(i)}$ la *i*-ème forme de bouche de la base d'apprentissage, z une forme de bouche candidate (ou pendant l'apprentissage, une forme $z^{(j)}$ issue de la base de données), σ^2 une variance fixée, et $\langle . \rangle$ la norme dont la formulation dépend du type de caractéristique utilisé, et sera donc explicitée dans les sections suivantes.

Dans nos expériences, nous considérons deux exemples d'apprentissage, l'un avec une décomposition de la vraisemblance en deux postures, *bouche fermée* et *bouche ouverte*, et l'autre avec une décomposition de la vraisemblance en trois postures, *bouche fermée*, *bouche ouverte*, et *sourire*. Dans le premier cas, les échantillons décrivant la posture *bouche ouverte* incluent ceux des postures *bouche ouverte* et *sourire* de la seconde décomposition.

Afin d'illustrer la classification effectuée par le SVM, nous accompagnerons nos commentaires de figures montrant les échantillons de la base d'apprentissage projetés dans un repère bidimensionnel, en effectuant une analyse en composantes principales et en conservant les deux premiers vecteurs propres (figure 6.3). Cette représentation est totalement indépendante du résultat de classification obtenu, c'est pourquoi nous ajoutons à celle-ci une information de couleurs, qui indique, via le résultat du SVM, à quelle classe un échantillon appartient. De plus, l'intensité de la couleur d'un échantillon est pondérée par sa probabilité d'appartenance, afin de mettre en évidence les échantillons proches de plusieurs classes (même si la projection en 2D de nos données n'a aucune valeur pour le classifieur SVM, qui effectue une séparation linéaire dans une plus grande dimension).



FIG. 6.3 – Projections en dimension 2, obtenues par une analyse en composantes principales, d'échantillons issus de la base d'apprentissage, en considérant (a) deux apparences, et (b) trois apparences.

6.4.2.1 Critères de distance entre histogrammes de couleurs

Comme premier critère caractérisant la probabilité d'une composante du modèle, nous considérons des histogrammes de couleurs. La fonction des caractéristiques γ correspond donc à l'histogramme d'une région, soit $\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos}) = \gamma(z) = h(z)$, ou comme nous avons pris l'habitude de le noter, $\gamma(z) = h(a_t)$, avec a_t l'ensemble des points de la région décrite par $(x_t, y_t, \theta_t, a_t)$ (cette écriture omet l'information issue de l'observation y_t^{Pos} dans un souci de simplification).

La distance utilisée entre deux histogrammes est celle de Bhattacharyya, ainsi la norme du noyau du SVM s'écrit :

$$\langle \gamma(z^{(i)}), \gamma(z) \rangle_{\text{Bhattacharyya}} = \langle h(z^{(i)}), h(z) \rangle_{\text{Bhattacharyya}}$$
$$= \left[1 - \sum_{u=1}^{U} \sqrt{h(z^{(i)}(u)) h(z(u))} \right]^{1/2}$$
(6.26)

avec $h(z^{(i)})$ le *i*-ème histogramme de couleurs HSV de la base d'apprentissage et h(z) un histogramme candidat.

La figure 6.4 illustre les projections des histogrammes issus de la base d'apprentissage, en considérant deux et trois postures, sachant que l'intensité et le diamètre de chaque échantillon sont proportionnels à la probabilité d'appartenance à sa classe. Dans cet apprentissage, toutes les étiquettes issues du SVM correspondent à celles fournies par la base d'apprentissage. Même si l'information est présentée est en 2*D*, nous pouvons constater que la probabilité d'appartenance des échantillons de la classe *bouche ouverte* diminue lorsqu'ils semblent proches de la classe *bouche fermée*. Il s'agit en réalité d'échantillons de bouches légèrement ouvertes, qui ont des histogrammes de couleurs proches de ceux de la classe *bouche fermée* (la figure 6.5 illustre les trois bouches dont les probabilités d'appartenance à la classe *bouche ouverte* sont les plus faibles). Nous pouvons également faire état de la faible dispersion spatiale de la classe *bouche fermée*, ce qui, encore une fois, paraît logique, puisque l'apparence en termes de couleurs d'une bouche fermée varie peu.



FIG. 6.4 – Projections en 2 dimensions, obtenues par une analyse en composantes principales, des histogrammes issus de la base d'apprentissage, en utilisant (a) deux postures, et (b) trois postures. Le diamètre ainsi que l'intensité de la couleur sont proportionnels aux probabilités d'appartenance des échantillons à la classe considérée.



FIG. 6.5 – Bouches de la base de données d'étiquette *bouche ouverte* dont les probabilités d'appartenance à leur propre classe sont les plus faibles.

6.4.2.2 Critères de distance euclidienne entre formes

Dans une deuxième expérience, nous caractérisons la probabilité d'une composante du modèle par une information de forme. Dans ce cas, la fonction de poids des postures ne dépend pas de l'observation, puisque l'on ne considère qu'un ensemble de points 2D, elle s'écrit donc par les points de contrôle de la forme a_t , soit $\varphi^j \left(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos}) \right) =$ $\varphi^j \left(\gamma(\mathbf{x}_t) \right) = \varphi^j \left(\gamma(z) \right) = \varphi^j(\mathbf{a}_t)$. En considérant une distance euclidienne entre formes, calculée sur les points de contrôle, la norme euclidienne s'écrit :

$$\langle \gamma(z^{(i)}), \gamma(z) \rangle_{\text{Euclidienne}} = \langle z^{(i)}, z \rangle_{\text{Euclidienne}}$$

$$= \left[\sum_{p=1}^{P} \left\| z^{(i)}(p) - z(p) \right\|_{2}^{2} \right]^{1/2}$$
(6.27)

avec $z^{(i)}$ la *i*-ème forme issue de la base de données, z une forme candidate et P le nombre de points de contrôle 2D.

La figure 6.6 illustre les projections des formes issues de la base d'apprentissage, en considérant deux et trois postures. Le SVM est construit en utilisant, pour le noyau, une distance euclidienne. Dans cet apprentissage également, les étiquettes issues du SVM correspondent à celles données par la base d'apprentissage. Par rapport à un critère d'histogramme de couleurs, les classes semblent plus dispersées, mais les probabilités obtenues par le SVM sont plus caractérisées : leurs probabilités d'appartenance semblent maximales, vis-à-vis des classes obtenues. Nous pouvons dire que ce critère semble discriminant, même si la projection en 2 dimensions est trompeuse, puisqu'il n'existe pas de séparation linéaire dans cet espace réduit. La proximité spatiale des classes *bouche ouverte* et *bouche fermée* semble logique, du fait des faibles différences de formes qu'il existe entre ces deux classes.



FIG. 6.6 – Projections en 2 dimensions, obtenues par une analyse en composantes principales, des formes issues de la base d'apprentissage, en utilisant (a) deux postures, et (b) trois postures. Le SVM est construit en utilisant une distance euclidienne. Le diamètre ainsi que l'intensité de la couleur sont proportionnels aux probabilités d'appartenance des échantillons à la classe considérée.

6.4.2.3 Critères de distance de Hausdorff entre formes

Comme dernière expérience, nous caractérisons la probabilité d'une posture par une information de forme, en considérant une distance de Hausdorff pour la norme du noyau

du SVM. Comme dans le cas de la distance euclidienne, la fonction de poids des postures ne dépend pas de l'observation, et les poids s'écrivent $\varphi^j \left(\gamma(\mathbf{x}_t, \mathbf{y}_t^{Pos})\right) = \varphi^j(\mathbf{a}_t)$. La distance de Hausdorff est donnée par :

$$\langle \gamma(z^{(i)}), \gamma(z) \rangle_{\text{Hausdorff}} = \max[d_H(\gamma(z), \gamma(z^{(i)})), d_H(\gamma(z^{(i)}), \gamma(z))]$$
(6.28)

avec d_H la distance de Hausdorff asymétrique définie par :

$$d_H(\gamma(z), \gamma(z^{(i)})) = \max_{p \in \gamma(z)} d_E(p, \gamma(z^{(i)}))$$
(6.29)

avec d_E la distance euclidienne, $\gamma(z)$ l'ensemble des points d'une région définie par une forme candidate, et p un point de $\gamma(z)$. Cette formulation est coûteuse en termes de temps de calculs, et dans le cadre du suivi, nous proposons d'en faire une approximation. Au lieu d'être calculée sur chaque paire de points des régions, nous considérons uniquement les paires de points de contrôle, donc, comme dans le cas de la forme avec distance euclidienne, z est un ensemble de points de contrôle, et $\gamma(z) = z$. Cette approximation semble adéquate, car dans la mesure où la bouche peut être considérée comme une forme pratiquement convexe, les distances maximales entre paires de points devraient être obtenues par des points de contrôle.

La figure 6.7 illustre les projections issues de la base d'apprentissage, en considérant deux et trois postures. Le SVM est construit en utilisant, pour le noyau, une distance de Hausdorff. Par rapport à une distance euclidienne, nous pouvons constater que nombre d'échantillons de la classe *bouche ouverte* se retrouvent, après la classification par SVM, dans la classe *bouche fermée* (un certain nombre d'échantillons sont passés de leur couleur verte étiquetée dans la base d'apprentissage à la couleur rouge). Le critère utilisé n'est peut-être pas assez discriminant, ou bien l'approximation réalisée dans le calcul de la distance de Hausdorff est trop grande. Une amélioration pourrait être portée en utilisant des distances de Hausdorff modifiées, tel que cela a été proposé dans [Simon et Tarel, 2006] dans le cadre de la détection et reconnaissance de panneaux de signalisation routière.



FIG. 6.7 – Projections en 2 dimensions obtenues par une analyse en composantes principales des formes issues de la base d'apprentissage, en utilisant (a) deux postures, et (b) trois postures. Le SVM est construit en utilisant une distance de Hausdorff. Le diamètre ainsi que l'intensité de la couleur sont proportionnels aux probabilités d'appartenance des échantillons à la classe considérée.

6.4.3 Résultats préliminaires

Les erreurs de suivi des trois expériences en utilisant un critère d'histogrammes de couleurs avec une distance de Bhattacharyya, de formes avec une distance euclidienne, et de formes avec une distance de Hausdorff, sont donnés sur la figure 6.8, en fonction du nombre de particules. Tout comme dans la section 4.4, les erreurs de suivi obtenues correspondent à 1 moins le ratio entre les points communs de la forme estimée et la vraie forme et le maximum des aires de ces deux objets (équation 4.19). Le bénéfice d'un modèle multi-apparences est clair, puisque, pour chacun des trois critères, la diminution de l'erreur entre une et deux postures est de l'ordre de 25%. Entre deux et trois postures, la différence se fait moins nette, certainement du fait que le partitionnement devient moins évident qu'avec deux postures.



FIG. 6.8 – Erreurs de suivi obtenues en utilisant comme critère de pondération une information (a) de formes avec une distance euclidienne (b) et de Hausdorff, et (c) de couleurs avec une distance entre histogrammes.

La comparaison entre critères de définition des probabilités des composantes du modèle est représentée dans la figure 6.9. Nous avons également illustré les résultats obtenus sans utiliser de classifieur SVM pour calculer les poids des postures, c'est-à-dire en les fixant de manière uniforme. Tout d'abord, nous pouvons remarquer que le critère d'histogramme de couleurs semble meilleur que tous les autres, sans doute grâce à la quantité d'information contenue dans un tel modèle, ce qui rend l'estimation des poids robuste, même lorsque l'environnement devient bruité (observations bruitées, mauvaise prédiction, etc.). Que les erreurs de suivi obtenues par une stratégie uniforme soient plus faibles que celles observées par les approches de formes semble un peu plus déconcertant. Plusieurs phénomènes peuvent expliquer cela. Premièrement, les résultats proposés dans ce chapitre n'en sont qu'à un stade préliminaire, le classifieur SVM ayant été appris avec un jeu de paramètre quelconque, or un apprentissage SVM nécessite souvent un choix heuristique du jeu de paramètres afin d'optimiser les résultats. Deuxièmement, les bonnes performances d'une loi uniforme sont en fait prévisibles, grâce à la nature même d'un filtre particulaire. Dans la mesure où celui-ci confronte les particules entre elles, une particule ayant obtenu une forte vraisemblance sur au moins une des deux (ou trois) postures sera toujours avantagée par rapport à celle qui n'en obtient aucune. Enfin, et sur ce point il y a certainement un travail particulier à effectuer, le SVM fournit très généralement des probabilités assez marquées, et donc pas assez nuancées, ce qui peut poser problème dans les phases de transition d'apparences. Sur cet aspect, une loi uniforme permet d'opérer des changements d'apparence graduels. Ce dernier point nous incite a, peut-être, retravailler le modèle en intégrant l'indice de la posture adéquate dans le filtre particulaire. Ainsi, nous pourrions définir une loi d'évolution de celle-ci, qui permettrait d'effectuer des changements progressifs, en combinant à cela une fonction d'importance qui correspond à notre fonction de poids actuelle.



FIG. 6.9 – Erreurs de suivi obtenues en utilisant différentes caractéristiques pour pondérer les vraisemblances, avec (a) deux postures (*bouche fermée* et *bouche ouverte*), et (b) avec trois postures (*bouche fermée, bouche ouverte* et *sourire*).

La figure 6.10 présente des images de résultats de suivi obtenus avec une, deux, et trois classes. La bouche estimée, en bleu, est l'espérance de Monte-Carlo. Le critère de poids utilisé est celui d'histogramme de couleurs, qui comme nous l'avons vu, fournit les meilleurs résultats. La différence entre les résultats obtenus avec une et deux/trois postures est nette, comme nous pouvons le constater sur les images 2 à 6. Entre deux et trois postures, les estimations des images 2 à 6 présentent également des différences, même si elles restent moins flagrantes. En particulier, l'orientation de la bouche estimée dans l'image 5 avec deux postures est clairement moins bonne que celle avec trois.

6.5 Conclusion

Nous avons proposé dans ce chapitre une densité de probabilité de vraisemblance à modèles multiples. L'originalité de ce travail est double : d'une part, chaque modèle est lié à une apparence de l'objet, problématique qui n'est jamais traitée explicitement. Deuxièmement, la mise en œuvre d'un tel modèle décompose la vraisemblance sous la forme d'une densité de mélange, dont les poids sont déterminés par rapport à l'état et à l'observation du temps t. Ce conditionnement permet d'adapter les poids en fonction des données et des particules. Cette problématique, qui fait partie de notre contribution, permet d'entrevoir nombre d'applications, et d'améliorer la robustesse du suivi en utilisant de manière originale plusieurs modalités, les poids des particules étant totalement indépendants de la vraisemblance considérée.

Nous avons proposé une application du suivi de bouche. Pour calculer les poids des composantes, que nous appelons postures, nous utilisons un SVM qui se charge de séparer, hors ligne, les caractéristiques considérées. Par rapport à un modèle n'utilisant pas de



FIG. 6.10 – Résultats obtenus avec (a) une, (b) deux, et (c) trois postures, dans une séquence de tests de 300 images. La forme estimée est représentée en bleue.

concept multi-composantes, l'amélioration est indéniable. En utilisant un critère de définition des poids des postures avec un histogramme de couleurs, nous avons obtenu des résultats satisfaisants, supérieurs à ceux obtenus avec des poids uniformes. Cependant, avec les critères de forme, nous avons constaté que l'apport n'est pas évident.

Ce travail préliminaire permet d'envisager plusieurs perspectives. Premièrement, multiplier les expériences (suivi de visage, d'objets quelconques, etc.), et les critères de poids utilisés (texture, points d'intérêts, etc.). Concernant la méthode d'apprentissage (qui rappelons-le, n'est pas obligatoire, mais est générale, ce qui permet de considérer des critères complexes), il serait sans doute intéressant de se tourner vers des méthodes non supervisées, de manière à ne pas spécifier les étiquettes, et laisser le classifieur trouver les ensembles les plus discriminants. Le modèle pourrait également subir des changements, en intégrant, si cela s'avère efficace, l'indice de la posture considérée dans le filtre particulaire. Ainsi, cela laisserait la possibilité de modéliser de manière souple la transition de l'indice de la posture, et la vraisemblance serait alors définie par l'indice considéré, à l'instar du modèle flou que nous avons défini dans le chapitre **3**.

l Chapitre

Conclusion et perspectives

7.1 Récapitulatif des contributions

Une des principales contributions de la thèse est l'intégration d'informations de haut niveau pour guider de manière efficace le suivi d'objet(s) par filtrage particulaire, réalisée en manipulant des informations spatiales floues. Dans un cadre mono-objet, ces informations peuvent être structurelles ou de forme, et permettent de représenter l'évolution de l'état de manière souple et générique, ce qui la rend robuste à diverses situations de suivi. Dans un contexte multi-objets, nous avons également intégré des relations spatiales entre les objets, ce qui permet de modéliser de manière intuitive des interactions complexes. D'autres contributions, en marge du fil conducteur de cette thèse, ont également été proposées. L'ensemble est discuté dans cette section.

7.1.1 Intégration d'informations floues pour le suivi d'objet

Nous avons tout d'abord proposé la modélisation d'informations spatiales floues dans le chapitre 2. Les informations peuvent être de deux types : structurelles d'une part, ce qui permet de définir des relations spatiales d'un objet entre deux instants (*à l'est de, orienté vers,* etc.), et des informations telles que les formes floues d'autre part, qui permettent de décrire la forme d'un objet ainsi que son évolution spatio-temporelle de manière implicite (*bouche ouverte, bouche fermée*, etc.).

Le cadre formel unifié que nous avons établi facilite l'intégration d'informations floues dans le filtre particulaire proposé dans le chapitre **3**. En les considérant comme des événements flous, les informations spatiales sont d'abord introduites dans un contexte probabiliste. Le partitionnement flou de l'espace des paramètres est décrit par un ensemble fini d'ensembles flous, définis par des fonctions d'appartenance. La sélection des ensembles flous décrivant l'état inconnu est introduite dans l'estimation du filtre particulaire, en s'apparentant à un modèle à sauts. La description des paramètres s'effectue alors à deux niveaux : une première information globale, qui désigne l'ensemble flou régissant le modèle dynamique du système, et une seconde information plus localisée, qui correspond aux paramètres du modèle, décrits par les ensembles flous précédemment établis.

La modélisation formelle d'informations spatiales floues, le cadre d'intégration, ainsi que son principe d'utilisation dans une problématique de suivi sont les éléments de cette première contribution. A cela, nous pouvons également ajouter, grâce à l'approche floue proposée, la définition d'une nouvelle fonction d'importance, qui dépend de la dernière observation et de la trajectoire de l'état. Il est bien connu que l'utilisation d'une fonction d'importance intégrant, entre autres, la dernière observation, améliore de manière conséquente la qualité de l'estimation fournie. En outre, l'utilisation de la trajectoire, principe qui n'avait à notre connaissance pas été exploité jusque ici, permet, en synthétisant cette information, d'intégrer une mémoire à long terme, et ainsi de lever l'ambiguïté sur des phénomènes spatio-temporels locaux.

Nous avons proposé une série d'expériences dans le chapitre 4. Nous avons tout d'abord étudié l'influence des paramètres sur notre modèle flou. Il en ressort une stabilité satisfaisante de notre approche vis-à-vis de ses paramètres, ce qui permet de les fixer de manière intuitive, sans connaissance *a priori* forte. Nous avons également conduit plusieurs tests, comparant l'approche proposée aux filtres SIR, AUX-PF, MMPF et AUX-MMPF. A travers plusieurs types de scénarios, linéaires comme erratiques, notre modèle flou, avec et sans fonction de proposition, fournit des résultats toujours supérieurs aux autres méthodes avec et sans fonction de proposition. L'analyse en termes de coûts de calcul a également montré qu'en modélisant convenablement la fonction de proposition de notre approche, le rapport entre temps de calcul et erreur obtenue est à l'avantage de nos modèles flous.

Enfin, nous avons mené une expérience visant à montrer la généricité de notre approche en modélisant des informations de formes, qui ont été apprises par un algorithme de Metropolis-Hasting. Formellement, nous avons défini un partitionnement de l'espace des formes, dont chaque élément correspond à une posture. Dans le cadre du suivi de lèvres, ces postures sont *bouche ouverte, bouche fermée* et *sourire*. Les résultats obtenus, que nous avons comparés à l'approche classique proposée par Isard et Blake [Isard et Blake, 1998a], montrent une supériorité de notre modèle, certainement grâce à ses propriétés de formes non contraintes, et une erreur de recouvrement de forme inférieure à 20%.

7.1.2 Intégration d'informations floues pour le suivi d'objets multiples

La première contribution sur la problématique multi-objets, developpée dans le chapitre 5, a été de proposer une amélioration du filtre particulaire avec échantillonnage partitionné, en ajoutant à l'estimation des états des objets celle de leur ordre de traitement. L'échantillonnage partitionné est une technique proposée par [MacCormick et Blake, 2000; MacCormick et Isard, 2000] qui permet d'attenuer le problème lié à la simulation par échantillonnage par importance en grande dimension, en partitionnant l'espace d'état en autant d'éléments qu'il y a d'objets, de manière à traiter chacun d'eux séquentiellement. Cependant, comme cela a été remarqué dans [Smith et Gatica-Perez, 2004], l'ordre de traitement séquentiel a une influence non négligeable sur les résultats obtenus, particulièrement dans des cas d'ambiguïté, telles que les occultations entre objets. Cela est dû à la nature séquentielle de l'algorithme, qui, en appauvrissant le nuage de particules à cause des rééchantillonnages successifs, diminue la diversité des états, spécialement ceux considérés au début du traitement. Notre stratégie d'échantillonnage partitionné à rang tente de résoudre ce problème, en estimant la séquence de traitement adéquate, et en diversifiant de ce fait les scénarios considérés.

Nous avons également proposé d'intégrer des contraintes spatiales floues dans le filtre particulaire multi-objets. Ces contraintes, exprimées comme des relations spatiales entre objets (d'orientation, d'alignement, de distance, etc.), permettent de restreindre l'ensemble des configurations spatiales possibles. Ainsi, selon le type d'application et de

séquence que l'on utilise, cela permet de facilement contraindre le positionnement des objets entre eux. Les deux contributions proposées dans ce chapitre sont indépendantes, ainsi les contraintes floues peuvent être utilisées dans un autre filtre particulaire de suivi, et même certainement dans un autre algorithme de suivi.

Diverses expériences ont été menées dans ce chapitre. Nous avons tout d'abord présenté une séquence de trois piétons pouvant s'occulter mutuellement. Utiliser un ordre de traitement adaptatif dans ce type de séquence s'avère particulièrement judicieux, puisque cela permet de considérer en premier les objets visibles, qui conditionnent ainsi l'occultation des autres objets. Notre algorithme d'échantillonnage partitionné à rang obtient alors de meilleurs résultats que la version classique. Nous avons ensuite analysé une séquence de fourmis, filmées de haut. Grâce au placement de la caméra, nous avons pu exploiter une hypothèse d'exclusion des fourmis, via une constrainte spatiale floue, qui a démontré, par les résultats obtenus, son intérêt par rapport à un suivi sans contrainte. Enfin, la dernière expérience est celle du suivi des doigts de la main, qui peuvent être partiellement visibles. Les caractéristiques anatomiques de la main ont été traduites dans notre formalisme flou par des contraintes d'alignement des doigts, d'angles, de distances, et d'exclusion. Cela permet, entre autres, de contraindre l'aspect global des résultats obtenus, et ainsi de suivre les doigts même lorsqu'ils ne sont que partiellement visibles. Une fois encore, les résultats comparés à ceux d'une méthode sans contrainte et à un échantillonnage partitionné classique avec contraintes se sont avérés concluants.

7.1.3 Utilisation de plusieurs modèles d'apparence pour le suivi d'objet

Le chapitre 6 traite de l'utilisation de plusieurs modèles d'apparence pour le suivi d'objet. Cette problématique n'avait, à notre connaissance, jamais été étudiée de cette manière, et offre plusieurs perspectives en termes de modélisation. Elle permet, d'une part, de modéliser les objets dont l'apparence change, que ce soit en termes de formes, de couleurs, ou encore de textures. Comme nous avons pu le constater dans nos expérimentations, cette stratégie s'avère efficace.

D'autre part, le modèle proposé considère la vraisemblance comme une densité de mélange, dont les poids dépendent de l'état, et de la dernière observation. Cette décomposition fait partie de notre contribution. Cela permet de définir des poids qui sont liés aux particules, ce qui se distingue des méthodes de la littérature qui utilisent une stratégie globale pour déterminer leurs valeurs. Ainsi, afin de montrer l'intérêt de cette décomposition, nous avons proposé d'estimer les poids via un classifieur SVM multi-classes.

Nous avons illustré ce travail par un suivi de la bouche, dont la posture pouvait passer de *bouche fermée* à *sourire*. En considérant soit un modèle classique d'une seule posture (*bouche*), soit une décomposition en deux postures (*bouche fermée* et *bouche ouverte*), soit une décomposition en trois postures (*bouche fermée, bouche ouverte* et *sourire*), les expériences ont montré qu'utiliser deux postures fournissait de meilleurs résultats qu'aucune, et que trois postures fournissaient d'encore meilleurs résultats. Cela paraît logique, puisque considérer plus de postures, et donc d'apparences, permet de mieux caractériser la représentation de l'objet. Pour définir les composantes du modèle, nous avons étudié l'utilisation de plusieurs caractéristiques : des histogrammes de couleurs avec une distance de Bhattacharyya, et des formes, en utilisant les distances euclidienne et de Hausdorff. Dans le cadre de nos tests, les résultats avec histogrammes sont sensiblemement meilleurs que ceux avec des formes, ce qui indique une meilleure caractérisation de l'objet avec ce critère. Toutefois, une étude plus en profondeur devrait être menée, afin de mieux cibler l'apport et le comportement de ces critères.

7.1.4 Bibliothèque générique de filtre particulaire pour le suivi d'objets

Une dernière contribution à ce travail de thèse concerne la mise œuvre en C++ d'une bibliothèque générique de filtres particulaires pour le suivi d'objets, disponible sur la page Internet http://perso.telecom-paristech.fr/~widynski/pfcpp/. La généricité vient de la programmation orientée objet qui permet de manipuler, définir, redéfinir, etc., des fonctions et méthodes, dans une structure spécifique, que l'on appelle bibliothèque. Dans le but d'intégrer facilement des algorithmes de traitement d'images, elle est interfacée avec la bibliothèque de traitement d'images OpenCV 1.0¹. Un aperçu de la bibliothèque de filtres particulaires est donné dans l'annexe C.

7.2 Perspectives

7.2.1 Intégration d'informations floues pour le suivi d'objet

Le modèle de manipulation d'informations spatiales floues unifié et générique a permis de définir un socle commun, à partir duquel il devient possible d'intégrer ces informations dans un contexte probabiliste, sans perte de généralité. Nous avons envisagé des combinaisons de stratégies de fusion différentes qui n'ont pas toutes été utilisées dans nos expériences. Par exemple, à aucun endroit dans nos expériences, nous n'avons utilisé de stratégies intra-classes conjonctives, de somme (à part pour la densité de simulation pour l'approximation de la loi *a priori* par algorithme de Metropolis-Hasting), ou encore de stratégies inter-classes exclusives. Une perspective intéressante serait donc de développer et d'approfondir ces différentes stratégies de fusion.

L'introduction d'informations spatiales floues dans un filtre particulaire s'est faite en combinant un modèle de paramètres à un modèle à sauts. L'utilisation de paramètres dynamiques permet de se rapprocher d'un modèle de suivi classique, uniquement probabiliste. Ainsi, le positionnement scientifique de cette première, et principale, contribution, a consisté à considérer des problématiques dont la dynamique réelle est inconnue, ce qui arrive fréquemment en pratique, et nécessite alors de poser des modèles de manière empirique. Dans ce domaine, les relations spatiales floues bénéficient de la souplesse de la représentation floue, ce qui rend le suivi robuste, tandis que les formes floues tirent avantage de l'aspect générique d'une modélisation par le flou. Une autre démarche aurait pu être choisie, et peut être mise en perspective de ce travail, qui aurait consisté à maintenir le niveau de raisonnement aux ensembles flous, sans faire intervenir de paramètres. Dans ce cadre, nous pourrions par exemple envisager d'utiliser des méthodes de fusion d'informations floues, afin de réduire l'imprécision autour de la localisation d'un objet.

En utilisant une variable aléatoire de sauts, pour spéficier les ensembles flous considérés, nous ajoutons à l'espace d'état un vecteur qui a autant de composantes qu'il y a de paramètres dynamiques. Comme nous l'avons vu, cela a théoriquement un impact sur la variance des poids des particules, et demande alors de garder plus d'échantillons pour une qualité d'estimation identique. Cependant, dans notre modèle graphique, les paramètres sont uniquement conditionnés aux variables de sauts, ce qui réduit l'impact de l'augmentation de la variable d'état en guidant assez précisément la simulation. Nous

¹http://sourceforge.net/projects/opencvlibrary/

pensons donc que le préjudice de l'introduction de cette variable discrète reste, en pratique, mineur.

Une perspective intéressante concerne la procédure d'approximation de la loi *a priori* par l'algorithme de Metropolis-Hasting, qui utilise actuellement une fonction de simulation naïve. Cela ralentit le processus d'approximation, et cela nous empêche en pratique de fixer des contraintes de formes trop fortes.

Sur les formes floues, des expériences pourraient être menées, dans le cadre de formes quelconques, et faisant intervenir d'autres distances entre formes. Multiplier les tests à ce sujet permettrait de mieux identifier et de confirmer l'apport de ce modèle flou de formes.

7.2.2 Intégration d'informations floues pour le suivi d'objets multiples

L'intégration de contraintes spatiales floues dans un filtre particulaire multi-objets offre au moins une perspective intéressante. Elle concerne le moment où l'information floue est utilisée dans le filtre. Dans un but de généralité, et conformément à la plupart des articles traitant du sujet, nous l'intégrons pendant la mise à jour des poids des particules. Cette méthodologie est générale puisqu'elle nécessite uniquement de savoir évaluer la densité en un point de l'espace. Simuler selon celle-ci est une opération nettement plus délicate, mais possède l'avantage d'être bien plus efficace. Un travail pourrait être mené afin de distinguer les cas où la connaissance analytique de cette densité peut être extraite, des cas où elle ne peut pas l'être, et donc adapter la mise en œuvre selon cette information.

L'objectif d'une méthode d'échantillonnage partitionné est de soulager le problème lié à la dimension de l'espace d'état en le décomposant en plusieurs éléments d'une seule partition. Ce qui peut sembler contradictoire dans notre approche d'échantillonnage partitionné à rang, c'est justement l'introduction d'un vecteur aléatoire discret, qui possède un nombre de composantes égal au nombre d'objets présents dans la scène. Ainsi, en gérant le problème de dimension, nous en ajoutons de nouvelles. Cela peut avoir une influence en pratique, même si nous ne l'avons pas constatée jusqu'ici, tant l'apport de notre stratégie de simulation semble efficace. Toutefois, d'autres expériences et un recul sur cette méthode permettraient certainement de mieux juger de son efficacité. La diversification des expériences ainsi qu'une étude plus appronfondie de la méthode proposée constituent des perspectives à court terme de ce travail.

Enfin, l'échantillonnage partitionné dynamique proposé dans [Smith et Gatica-Perez, 2004] devrait prochainement intervenir dans les expériences menées dans le chapitre 5, afin de comparer ses résultats à ceux de l'approche proposée, puisqu'il s'agit d'une autre méthode visant à gérer le phénomène d'appauvrissement des particules introduit par l'échantillonnage partitionné classique.

7.2.3 Utilisation de plusieurs modèles d'apparence pour le suivi d'objet

La dernière contribution proposée dans ce manuscrit, développée en fin de thèse, mérite d'être plus explorée. Les expériences doivent être améliorées, et diversifiées. Le modèle pourrait également connaître quelques changements, en intégrant, par exemple, la sélection de la posture à l'estimation du filtre particulaire.

Une perspective intéressante à ce travail serait de calquer le principe de composantes multiples au cas multi-caractéristiques et/ou multi-capteurs. Cela signifie que l'on agrémenterait d'un poids, ou indice, les informations d'un capteur, qui indiquerait la probabilité, ou la fiabilité de la modalité, comme cela a déjà été traité dans la littérature. Notre approche se démarquerait de celles existantes dans la manière de calculer cette quantité, en la conditionnant toujours par rapport à l'état et aux observations. Donc le principe serait le même que celui des postures multiples, les quantités seraient uniques et définies par les particules. Cela pourrait être particulièrement intéressant lorsque l'efficacité d'un capteur peut dépendre de l'état, ce qui n'a, à notre connaissance, pas été étudié jusqu'ici. Par exemple, avec une variable d'état estimant la position et la vélocité d'un objet, il serait intéressant de pondérer l'importance d'une caractéristique de flot optique par une information de couleurs, lorsque l'objet ne se déplace pas. Ou encore, dans un cadre multi-capteurs, nuancer la pertinence de l'information fournie par l'un d'eux selon la position ou l'état de l'objet peut également servir dans un grand nombre d'applications.

7.2.4 Autres perspectives

Comme nous l'avons constaté dans ce manuscrit, l'utilisation en pratique d'un filtre particulaire souffre de plusieurs problèmes, qui à ce jour restent ouverts. La limitation due à une dimension de l'espace d'état élevée en est une, et est généralement un frein à la mise en œuvre d'un algorithme de suivi multi-objets par filtre particulaire, ou plus généralement, aux problématiques faisant intervenir de grandes dimensions de l'espace d'état, telles que le suivi d'objets articulés, ou les problématiques de suivi 3*D*, de plus en plus à la mode.

La définition du modèle dynamique de l'état est très importante en pratique, puisqu'elle régit la propagation des particules, et donc conditionne le succès de l'algorithme. Dans des applications de suivi, il est fréquent d'avoir peu d'informations sur la loi de transition, et ainsi de considérer un modèle pauvre. L'étude de modèles adaptatifs, par exemple en utilisant des hyperparamètres du modèle dynamique, a peu été étudiée jusqu'ici. L'apprentissage, pendant le suivi, du modèle dynamique est également une perspective intéressante, qui reste encore peu developpée. D'autre part, la définition d'une fonction d'importance qui dépend de la dernière observation est pratiquement obligatoire pour mener à bien le suivi, et pourrait être encouragée en intégrant, ou en combinant, au filtre particulaire, des algorithmes de traitement d'images de détection, ou d'estimation de mouvement. Cette problématique a déjà été traitée dans la littérature, mais reste encore assez marginale.

Enfin, l'utilisation du filtre particulaire en traitement d'images se cantonne bien souvent aux problématiques de suivi. La segmentation par les contours est certainement une perspective intéressante, et comporte toutes les difficultés rencontrées par le filtre particulaire : détections et disparitions, forte dimension de l'espace d'état, multi-objets, maintien de la diversité des particules, etc.

7.3 Publications

- Nicolas Widynski, Séverine Dubuisson and Isabelle Bloch. *Integration of Fuzzy Spatial Information in Tracking Based on Particle Filtering*. IEEE Transactions on Systems, Man and Cybernetics SMC-B, 2010. A paraître.
- Nicolas Widynski, Emi Aldea, Séverine Dubuisson and Isabelle Bloch. *Object tracking based on particle filtering with multiple appearance models*. International Conference on Computer Vision Theory and Applications (VISAPP). Algarve, Portugal, Mars, 2011.

- Nicolas Widynski, Séverine Dubuisson and Isabelle Bloch. *Introducing Fuzzy Spatial Constraints in a Ranked Partitioned Sampling for Multi-Object Tracking*. Proceedings of the 6th International Symposium on Visual Computing. Las Vegas, Novembre, 2010.
- Nicolas Widynski, Séverine Dubuisson and Isabelle Bloch. *Particle filtering with fuzzy spatial relations for object tracking*. IEEE Image Processing Theory, Tools and Applications. IPTA'10, Paris, France, Juillet 2010. pp. 391-396.
- Nicolas Widynski, Séverine Dubuisson et Isabelle Bloch. *Intégration d'informations spatiales floues dans le filtre particulaire : application au suivi de formes*. Conférence en Reconnaissance des Formes et Intelligence Artificielle (RFIA). Caen, France, Janvier, 2010. (Cet article a reçu le prix du meilleur article de cette conférence dans la section Reconnaissance des Formes).
- Nicolas Widynski, Séverine Dubuisson et Isabelle Bloch. *Intégration de relations spatiales floues dans un filtre particulaire pour le suivi d'objets*. Actes du GRETSI. Dijon, France, Septembre, 2009.

7. CONCLUSION ET PERSPECTIVES

Annexe A

Compléments au chapitre 2 : preuve de la proposition 2.1

Soit $\mathcal{O} = [I_1^1, I_2^1] \times \cdots \times [I_1^R, I_2^R] \subset \mathbb{R}^R$ le domaine de la partition de dimension R. Soit $\forall r \in \{1, \ldots, R\}, m^r = (m_1^r, \ldots, m_{K^r}^r)$ les K^r nœuds fixés de l'intervalle $[I_1^r, I_2^r]$, tels que $m_1^r = I_1^r$ et $m_K^r = I_2^r$, avec $K^r \ge 2$, et $\forall k \ne K^r, m_{k+1}^r - m_k^r = h^r$. Soit $\forall (k^1, \ldots, k^R) \in \{1, \ldots, K^1\} \times \cdots \times \{1, \ldots, K^R\}, \forall x \in \mathcal{O}$, la fonction d'appartenance jointe $\mu_{k^1, \ldots, k^R}(x) = \prod_{r=1}^R \mu_{k^r}^r(x^r)$. Soit $\forall r \in \{1, \ldots, R\}$, la famille des ensembles flous $\{\mathcal{A}_{k^r}^r\}_{k^r=1}^{K^r}$ identifiés par leurs fonctions d'appartenance $\{\mu_{k^r}^r\}_{k^r=1}^{K^r}$ formant une partition uniforme floue forte sur le domaine $[I_1^r, I_2^r]$.

Montrons que la famille des ensembles flous $\{A_{k^1,...,k^R}\}$ identifiés par leurs fonctions d'appartenance $\{\mu_{k^1,...,k^R}\}$ forme une partition uniforme floue forte sur \mathcal{O} .

Preuve de la proposition 2.1 *La preuve est directe, il suffit de vérifier les propriétés énoncées dans la définition* **2.4** *:*

1. $\forall (k^1, \ldots, k^R) \in \{1, \ldots, K^1\} \times \cdots \times \{1, \ldots, K^R\},\$

$$\mu_{k^{1},\dots,k^{R}}(m_{k^{1}}^{1},\dots,m_{k^{R}}^{R}) = \prod_{r=1}^{R} \mu_{k^{r}}^{r}(m_{k^{r}}^{r})$$
$$= 1 \times \dots \times 1 = 1$$

2. $\forall (k^1, \dots, k^R) \in \{1, \dots, K^1\} \times \dots \times \{1, \dots, K^R\}, \forall x \notin [m_{k^1-1}^1, m_{k^1+1}^1] \times \dots \times [m_{k^R-1}^R, m_{k^R+1}^R],$

$$\mu_{k^1,\dots,k^R}(x) = \prod_{r=1}^R \mu_{k^r}^r(x^r)$$
$$= 0 \times \dots \times 0 = 0$$

3. $\forall r \in (1, ..., R), \forall k^r \in \{1, ..., K^r\}, \mu_{k^r}^r(x) \text{ a une croissance monotone sur } [m_{k^r-1}^r, m_{k^r}^r]$ et une décroissance monotone sur $[m_{k^r}^r, m_{k^r+1}^r]$ 4. $\forall x \in \mathcal{O}$,

$$\sum_{k^{1}=1}^{K^{1}} \cdots \sum_{k^{R}=1}^{K^{R}} \mu_{k^{1},\dots,k^{R}}(x) = \sum_{k^{1}=1}^{K^{1}} \cdots \sum_{k^{R}=1}^{K^{R}} \prod_{r=1}^{R} \mu_{k^{r}}^{r}(x^{r})$$
$$= \sum_{k^{1}=1}^{K^{1}} \mu_{k^{1}}^{1}(x^{1}) \times \cdots \times \sum_{k^{R}=1}^{K^{R}} \mu_{k^{R}}^{R}(x^{R})$$
$$= \prod_{r=1}^{R} \sum_{k^{r}=1}^{K^{r}} \mu_{k^{r}}^{r}(x^{r})$$
$$= 1 \times \cdots \times 1 = 1$$

5. $\forall (k^1, \dots, k^R) \in \{2, \dots, K^1 - 1\} \times \dots \times \{2, \dots, K^R - 1\}, \forall x \in [0, h^1] \times \dots \times [0, h^R],$

$$\mu_{k^{1},\dots,k^{R}}(m_{k^{1}}^{1}-x^{1},\dots,m_{k^{R}}^{R}-x^{R}) = \prod_{r=1}^{R} \mu_{k^{r}}^{r}(m_{k^{1}}^{r}-x^{r})$$
$$= \prod_{r=1}^{R} \mu_{k^{r}}^{r}(m_{k^{1}}^{r}+x^{r})$$
$$= \mu_{k^{1},\dots,k^{R}}(m_{k^{1}}^{1}+x^{1},\dots,m_{k^{R}}^{R}+x^{R})$$

6. $\forall (k^1, \dots, k^R) \in \{2, \dots, K^1 - 1\} \times \dots \times \{2, \dots, K^R - 1\}, \forall x \in [m_{k^1}^1, m_{k^1 + 1}^1] \times \dots \times [m_{k^R}^R, m_{k^R + 1}^R],$

$$\begin{split} \mu_{k^{1},\dots,k^{R}}(x) &= \prod_{r=1}^{R} \mu_{k^{r}}^{r}(x^{r}) \\ &= \prod_{r=1}^{R} \mu_{k^{r}-1}^{r}(x^{r}-h^{r}) \\ &= \mu_{k^{1}-1,\dots,k^{R}-1}(x-h) \end{split}$$

7. $\forall (k^1, \dots, k^R) \in \{2, \dots, K^1 - 1\} \times \dots \times \{2, \dots, K^R - 1\}, \forall x \in [m_{k^1 - 1}^1, m_{k^1}^1] \times \dots \times [m_{k^R - 1}^R, m_{k^R}^R],$

$$\mu_{k^{1},\dots,k^{R}}(x) = \prod_{r=1}^{R} \mu_{k^{r}}^{r}(x^{r})$$
$$= \prod_{r=1}^{R} \mu_{k^{r}+1}^{r}(x^{r}+h^{r})$$
$$= \mu_{k^{1}+1,\dots,k^{R}+1}(x+h)$$

Vérifions maintenant que si la propriété énoncée dans l'équation 2.5 est vraie pour toutes les familles de fonctions d'appartenance monodimensionnelles $(\mu_{kr}^r)_{k^r=1}^{K^r} (\forall r \in \{1, \ldots, R\})$, alors elle vraie dans le cas multidimensionnel, avec $\mu_{k^1,\ldots,k^R} = \prod_{r=1}^R \mu_{k^r}^r$:

$$\begin{aligned} 8. \ \forall (k^1, l^1) \times \cdots \times (k^R, l^R) &\in \{1, \dots, K^1\}^2 \times \cdots \times \{1, \dots, K^R\}^2, \\ \int_{\mathcal{O}} \mu_{k^1, \dots, k^R}(x) \, dx &= \int_{I_1^1}^{I_2^1} \cdots \int_{I_1^R}^{I_2^R} \prod_{r=1}^R \mu_{k^r}^r(x^r) \, dx^1 \cdots dx^R \\ &= \prod_{r=1}^R \int_{I_1^r}^{I_2^r} \mu_{k^r}^r(x^r) \, dx^r \\ &= \prod_{r=1}^R \int_{I_1^1}^{I_2^r} \dots \int_{I_1^R}^{I_2^R} \prod_{r=1}^R \mu_{l^r}^r(x^r) \, dx^1 \cdots dx^R \\ &= \int_{\mathcal{O}} \mu_{l^1, \dots, l^R}(x) \, dx \end{aligned}$$

Annexe B

Compléments au chapitre 4 : expériences supplémentaires

B.1 Influence des paramètres : formes des fonctions d'appartenance et paramètre ϵ

Nous étudions dans cette section l'influence des formes des fonctions d'appartenance (triangulaire, trapézoïdale et rectangulaire) et du paramètre ϵ sur les résultats obtenus. La figure B.1 présente les erreurs obtenues en fonction du nombre de particules. Nous pouvons y constater la supériorité des modèles trapézoïdal et triangulaire sur le modèle rectangulaire dans le cas où $\epsilon \leq 0.1$. La figure B.2 présente les erreurs obtenues en fonction du degré minimal d'intersection, et montre ainsi la stabilité des formes trapézoïdale et triangulaire. L'utilisation d'une fonction triangulaire et d'une valeur de paramètre $\epsilon = 0.1$ fournit les résultats à la fois les plus probants et stables. Cela est particulièrement visible dans la séquence linéaire, dans laquelle l'utilisation d'un modèle de transition souple semble évident.

B.2 Séquence synthétique : modèle dynamique polaire

Nous considérons ici un exemple synthétique de suivi, où un objet est défini par le vecteur d'état $x_t = [\mathbf{x}_t \quad \mathbf{y}_t]^T$ de ses coordonnées spatiales, ainsi que par ses paramètres dynamiques exprimés en coordonnées polaires $\mathbf{r}_t = [\varphi_t \quad \mathbf{d}_t]^T$. Soient $\mathbf{c}_t^1 \in$ $\{E, NE, N, NO, O, SO, S, SE\}$ et $\mathbf{c}_t^2 \in \{proche, à moyenne distance, loin\}$ les valeurs des variables linguistiques. Nous considérons le système suivant :

$$\mathbf{x}_{t} = f_{t}(\mathbf{x}_{t-1}, \mathbf{r}_{t}, \mathbf{v}_{t})$$

$$= \begin{bmatrix} \mathbf{x}_{t-1} + \Delta t \, \mathrm{d}_{t} \cos(\varphi_{t}) + \frac{\Delta t^{2}}{2} \, \mathbf{v}_{t}^{1} \\ \mathbf{y}_{t-1} + \Delta t \, \mathrm{d}_{t} \sin(\varphi_{t}) + \frac{\Delta t^{2}}{2} \, \mathbf{v}_{t}^{2} \end{bmatrix}$$
(B.1)

$$\mathbf{r}_t = \mathbf{r}_{t-1} + \Delta t \, \mathbf{I}_{2 \times 2} \, [\mathbf{v}_t^2 \, \mathbf{v}_t^3]^T \tag{B.2}$$

$$\mathbf{y}_t = h_t(\mathbf{x}_t, \mathbf{r}_t, \mathbf{w}_t) = \mathbf{x}_t + \mathbf{w}_t \tag{B.3}$$



FIG. B.1 – Formes des fonctions d'appartenance : erreurs obtenues en fonction du nombre de particules, de la forme des fonctions floues (triangulaire, trapézoïdale et rectangulaire) et avec un degré minimal d'intersection entre deux classes ϵ fixé. (a,d,g,j,m,p) séquence linéaire (scénario synthéthique 1) ; (b,e,h,k,n,p) séquence erratique (scénario synthéthique 2) et ; (c,f,i,l,o,r) séquence classique (scénario synthéthique 3). (a-c) $\epsilon = 0$, (d-e) $\epsilon = 0.05$, (g-i) $\epsilon = 0.1$, (j-l) $\epsilon = 0.2$, (m-o) $\epsilon = 0.5$ et (p-r) $\epsilon = 1$.



FIG. B.2 – Formes des fonctions d'appartenance : erreurs obtenues en fonction du degré minimal d'intersection entre deux classes ϵ , de la forme des fonctions floues (triangulaire, trapézoïdale et rectangulaire) et avec un nombre de particules N fixé. (a,d,g) séquence linéaire (scénario synthéthique 1); (b,e,h) séquence erratique (scénario synthéthique 2) et; (c,f,i) séquence classique (scénario synthéthique 3). (a-c) N = 5, (d-e) N = 20 et (g-i) N = 50.

avec

$$\begin{aligned} \mathbf{v}_t^1, \mathbf{v}_t^2) &\sim & \mathcal{N}(\mathbf{0}_{2 \times 1}, 4^2 \mathbf{I}_{2 \times 2}) \\ \mathbf{v}_t^3 &\sim & \mathcal{N}(0, 1) \\ \mathbf{v}_t^4 &\sim & \mathcal{N}(0, 6^2) \\ \mathbf{w}_t &\sim & \mathcal{N}(\mathbf{0}_{2 \times 1}, 5^2 \mathbf{I}_{2 \times 2}) \end{aligned}$$
(B.4)

Afin d'illustrer le potentiel de notre approche, une trajectoire arbitraire d'un objet ponctuel a été considérée. Les observations sont générées à partir de cette trajectoire et selon le processus de mesure décrit dans l'équation B.3. La figure B.3 (a) compare les trajectoires obtenues pour une observation donnée. La trajectoire obtenue par la méthode proposée est comparée à celles obtenues par un filtre SIR et un filtre particulaire auxiliaire (AUX-PF), les filtres à modèles multiples MMPF et AUX-MMPF n'ayant pas été étudiés dans ces expériences préliminaires.

A chaque instant, l'état estimé correspond à l'espérance de \mathbf{x}_t^n . Les erreurs moyennes de ces positions sont illustrées sur la figure B.3 (b). Le tableau B.1 fournit l'erreur moyenne quadratique du suivi obtenu avec notre méthode, le filtre SIR, et l'AUX-PF, en considérant un nombre croissant de particules, et en moyennant les résultats obtenus avec 100 simulations d'observations. Pour le modèle proposé, nous avons utilisé $\beta^1 = 0.1, \beta^2 = 0.8$ et $\beta^3 = 0.1$, afin d'attirer les particules vers les zones de forte vraisemblance. Nous avons en outre fixé $\varepsilon = 0.1$ afin d'autoriser avec une faible probabilité des sauts conséquents pour la loi de transition des indices des fonctions floues. Des tests supplémentaires ont été menés en utilisant des formes de fonctions d'appartenance différentes, en allant des fonctions indicatrices binaires aux fonctions d'appartenance triangulaires. Dans tous nos tests, la dissimilarité des erreurs obtenues en utilisant différentes formes n'a pas été significative. Notons que la méthode proposée a pour principal avantage d'introduire une fonction d'importance pertinente, ce qui explique la convergence des filtres lorsque le nombre de particules est élevé. Cependant, avec un faible nombre de particules, l'approche fournit de manière notable de meilleurs résultats. Enfin, l'introduction d'une nouvelle information induit un coût additionnel par particule, qui est néanmoins compensé par le faible nombre de particules nécessaires à l'obtention d'une erreur moyenne quadratique acceptable.

Modèles	RMSE							
	5	20	50	100	500	2000		
PFS-PF	10.4	7.5	7.0	6.8	6.5	6.3		
SIR	33.8	11.0	7.6	7.1	6.8	6.5		
AUX-PF	29.4	9.6	7.7	7.4	7.2	6.8		

TAB. B.1 – Erreurs moyennes quadratiques d'estimation obtenues en considérant 100 simulations du processus d'observation, en utilisant N = 5, 20, 50, 100, 500 ou 2000 particules.

B.3 Séquence réelle : voiturette

Nous considérons dans cette section un problème de suivi d'objet dans une séquence vidéo. Cette séquence représente une voiturette noire avec une dynamique complexe,



FIG. B.3 – (a) Comparaison des trajectoires de suivi estimées obtenues avec N = 20 particules. (b) Erreurs associées.

sur un fond de parquet en bois. L'objectif est d'estimer la position du rectangle entourant l'objet. Nous modélisons ce problème avec $x_t = [\mathbf{x}_t \ y_t]^T$ le vecteur position de l'objet, et avec $\mathbf{r}_t = [\dot{\mathbf{x}}_t \ \dot{\mathbf{x}}_t \ \theta_t]^T$ le vecteur de paramètres. Pour $(\dot{\mathbf{x}}_t, \dot{\mathbf{y}}_t)$, nous considérons neuf valeurs linguistiques $\mathbf{c}_t^1 \in \{NO, N, NE, O, Centre, E, SO, S, SE\}$, et quatre pour l'orientation de l'objet $\mathbf{c}_t^2 \in \{N, NE, E, SE\}$. Le système à vélocité constante s'écrit :

$$\mathbf{x}_{t} = f_{t}(\mathbf{x}_{t-1}, \mathbf{r}_{t}, \mathbf{v}_{t})$$

$$= \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \end{bmatrix} \begin{bmatrix} \mathbf{x}_{t-1} \\ \mathbf{r}_{t} \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^{2}}{2} & 0 \\ 0 & \frac{\Delta t^{2}}{2} \end{bmatrix} \begin{bmatrix} \mathbf{v}_{t}^{1} \\ \mathbf{v}_{t}^{2} \end{bmatrix}$$
(B.5)

$$\mathbf{r}_t = \mathbf{r}_{t-1} + \Delta t \, \mathbf{I}_{3 \times 3} \, \mathbf{v}_t \tag{B.6}$$

avec

$$\begin{aligned} \mathbf{v}_t^1 \ \mathbf{v}_t^2 \end{bmatrix}^T &\sim \mathcal{N}(\mathbf{0}_{2\times 1}, 4^2 \mathbf{I}_{2\times 2}) \\ \mathbf{v}_t^3 &\sim \mathcal{N}(0, 0.3^2) \end{aligned}$$
(B.7)

L'objet est représenté par un rectangle dont les dimensions sont fixées. La vraisemblance utilise une distance entre un histogramme de couleurs d'un modèle et un histogramme candidat (voir par exemple [Nummiaro et al., 2002; Pérez et al., 2002] pour plus de détails, ou la section 4.1.2.2). Le tableau B.2 fournit les erreurs quadratiques moyennes des positions de la voiture, obtenues par notre modèle (PFS-PF), le filtre SIR, et l'AUX-PF.

L'approche proposée fournit de meilleurs résultats, en particulier pour un faible nombre de particules. De plus les résultats obtenus par le modèle proposé varient moins que ceux des modèles SIR et AUX-PF. Les erreurs sont illustrées sur la figure B.4. La figure B.5 compare les rectangles estimés obtenus par les trois méthodes. Celles obtenues par notre approche sont plus précises que celles obtenues par le filtre SIR et l'AUX-PF. Enfin, le modèle proposé semble plus robuste aux changements soudains de la trajectoire de l'objet et fournit une estimation plus stable de l'orientation de l'objet.

Modèles	Erreur moyenne quadratique							
	5	20	50	100	500	2000		
PFS-PF	21.7	12.8	10.4	9.1	7.7	7.6		
SIR	145.2	24.5	13.1	10.0	8.0	7.8		
AUX-PF	95.7	17.2	14.5	12.4	10.9	8.6		

TAB. B.2 – Erreurs moyennes quadratiques d'estimation obtenues moyennées sur 100 lancés, en utilisant N = 5, 20, 50, 100, 500 ou 2000 particules.



FIG. B.4 – Comparaison des erreurs obtenues pour N = 20 particules.



FIG. B.5 – Images des estimations obtenues pour N = 20 particules aux temps 7, 14, 24, 34, 43, 54, 67 et 76. Bleu : PFS-PF, vert : SIR, et rouge : AUX-PF.

Annexe

Bibliothèque de filtrage particulaire

Nous avons développé, dans le cadre de cette thèse, une bibliothèque générique de filtrage particulaire, écrite en C++. La généricité mise en œuvre permet de facilement adapter le code à tous types de modélisation et de données. Cependant, nous avons spécialement travaillé à l'écriture de filtres particulaires dédiés au suivi en traitement d'images, c'est pourquoi la bibliothèque proposée est interfacée avec la bibliothèque libre de traitement d'images OpenCV $1.0^{1,2}$.

Bien que tout le code écrit durant cette thèse ait été fait dans cette bibliothèque, la version disponible sur le site http://perso.telecom-paristech.fr/~widynski/ pfcpp/ n'intègre pour l'instant que les algorithmes de filtrage particulaire de base, mais permet néanmoins d'utiliser toutes les fonctionnalités proposées par la bibliothèque. Celle-ci a été spécialement étudiée afin de permettre à n'importe quel utilisateur de développer facilement son propre filtre particulaire.

L'objectif de cette annexe est de donner une rapide introduction à la conception de la bibliothèque de filtrage particulaire, ainsi qu'à son utilisation. La documentation, qui n'est pas encore terminée mais permet toutefois d'avoir une vue globale de la bibliothèque, est disponible sur le site http://perso.telecom-paristech.fr/ ~widynski/pfcpp/doc/.

C.1 Programmation de la bibliothèque

La bibliothèque est scindée en trois parties. La première régit tout le fonctionnement interne des algorithmes de filtrage particulaire (allocation, (ré)échantillonnage, sauvegarde de l'historique, de l'erreur, ...), des procédés d'entrées et sorties (entrées/sorties des flux d'images (séquence d'images, vidéo, ou webcam), lecture/écriture des fichiers), des états du filtres (eux-mêmes décomposés en trois parties : position, représentation, auxiliaires). Par exemple, la classe « représentation » définit une hiérarchie dans laquelle se trouvent les classes « ensemble ouvert » et « ensemble fermé ». Cette dernière définit la représentation des objets 2*D*, de types rectangulaires, elliptiques, ou de formes quelconques. La seconde structure recense tous les outils facilitant la programmation des filtres de suivi : algorithmes d'approximation MCMC, simulations de lois analytiques,

¹http://sourceforge.net/projects/opencvlibrary/

²Cela signifie que la bibliothèque de filtrage particulaire nécessite l'installation d'OpenCV. Cette contrainte devrait être levée dans une future version de la bibliothèque.

gestion des formes B-Splines, filtre de Kalman, relations spatiales floues, lois d'évolution, vraisemblances, histogrammes, options de la ligne de commande... Enfin, la dernière composante est l'unique interface de l'utilisateur. Elle lui laisse donc la possibilité de surcharger certaines fonctionnalités, en lui donnant au minimum trois fonctions à définir : initialisation du filtre, évolution et calcul de la vraisemblance. L'organisation de la bibliothèque est schématisée sur la figure C.1.



FIG. C.1 – Représentation schématique de la bibliothèque de filtrage particulaire. Une connexion avec un losange indique une relation d'agrégation, une flèche une relation d'héritage. Une classe abstraite (non instanciable) est représentée par un rectangle, une classe concrète par une forme arrondie. La classe PF gère l'ensemble de la structure : elle définit un ensemble de particules, elles-mêmes définissant un ensemble d'états (un état est une composante du vecteur, c'est-à-dire, dans un cadre multi-objets, un état représente un objet). L'état agrège trois concepts orthogonaux, que sont sa position, sa représentation (facultative, telles qu'une ellipse, une forme ouverte, etc.), et ses données auxiliaires (facultatives, telles que la visibilité d'un objet, sa catégorie, etc.). D'un point de vue utilisateur, seule la classe User est à développer. Elle permet de définir le comportement d'une particule à l'initialisation, la propagation et le calcul de la vraisemblance. Sa définition permet de sélectionner le triplet (position, représentation, autre) adéquat, tout en laissant la possibilité de redéfinir un élément, ou plusieurs, de ce triplet.

Comme nous l'avons déja signalé, un état est décrit par trois attributs, la position, la représentation, et les données auxiliaires. La position contient la localisation de l'état, ou en 2D, de l'objet, ainsi que des informations dynamiques : vecteurs vitesse et accélération. Chacune de ses informations peut être 1D, 2D, ou 3D.

La représentation définit une interface dont hérite directement la représentation 2D, elle-même décrivant une hiérarchie avec les classes :

- ellipse,
- rectangle,
- ensemble de points 2D, qui est un type abstrait, mais dont héritent les classes :
 objet,
 - forme ouverte.

Ces représentations permettent de facilement construire un algorithme de suivi dans des séquences d'images. Bien sûr, il est possible de définir un nouveau type, en construisant

une classe qui hérite directement d'une des classes sus-mentionnées. Définir une représentation est toutefois facultative.

Enfin la classe de données auxiliaires Other permet d'ajouter à l'estimation du filtre particulaire tout ce qui n'a pas été prévu dans les deux premières. Par exemple, elle peut contenir la visibilité d'un objet, sa catégorie, ou encore les hyperparamètres de sa dynamique.

Afin de faciliter la mise en œuvre d'un filtre, des outils sont également proposés, dont un certain nombre ont déjà été mentionnés. Ces outils sont intégrès à la bibliothèque proposée, ce qui rend l'écriture des programmes plus fluide, comme nous le verrons dans la section suivante. Dans la version actuellement proposée en téléchargement, les algorithmes boostrap, Condensation, SIS (algorithme 1.8), SIR (algorithme 1.10) et AUX-PF (algorithme 1.11), sont proposés.

Enfin, une combinaison, bien que restant principalement anecdotique dans notre bibliothèque, entre l'héritage par objet classique et l'utilisation de la méta-programmation permet de définir de manière efficace les opérations sur les types. Pour plus d'informations, le lecteur peut se référer à la bibliothèque générique de traitement d'images Olena³, qui allie généricité et performances de temps de calculs grâce à l'utilisation de structures d'héritages statiques.

C.2 Exemple

Nous donnons dans cette section un exemple complet de fichier C++ utilisant notre bibliothèque. La première étape consiste à définir, si besoin, une classe Other, qui contient les données auxiliaires du filtre. Dans cet exemple « jouet », nous considérons un modèle à sauts, il convient donc d'estimer l'indice du modèle qui gouverne l'évolution de l'état.

classe Other

```
// Définition d'une classe OtherSpl, qui hérite directement de la classe Other.
class OtherSpl : public Other<OtherSpl> {
public:
   OtherSpl() : dynamic_model_() {}
   OtherSpl(const OtherSpl& o) : Other<OtherSpl>(o) {}
   // Surcharge des opérateurs + et *.
   OtherSpl operator+(const OtherSpl& rhs) const { return rhs; }
   OtherSpl operator*(const double& s) const { return *this; }
   ~OtherSpl() {}
   // Nous considérons un exemple jouet avec un modèle de la dynamique changeant.
   // Ainsi, il convient de rajouter l'indice de la dynamique à l'estimation de l'état.
```

```
int dynamic_model_;
```

};

Nous définissons ensuite la classe utilisateur, que nous appelons ici UserTest. Elle a pour but de définir le comportement du filtre pendant les trois étapes essentielles d'un filtre particulaire : l'initialisation, la prédiction et la mise à jour des poids. Comme vraisemblance, nous considérons ici une distance entre des histogrammes de couleurs, l'un

³http://olena.lrde.epita.fr/

provenant du candidat, l'autre d'un modèle de référence, comme cela a notamment été décrit dans la section 4.1.2.

classe User

// Définition de la classe que doit spécifier l'utilisateur. Elle permet de définir le

// comportement des particules à l'initialisation, pendant la propagation, et pendant la // vraisemblance.

```
class UserTest : public User<Position2d , Ellipse , OtherSpl> {
  public :
```

```
UserTest() : User<Position2d , Ellipse , OtherSpl>() {}
```

```
// Initialisation d'une composante. id désigne la id-ème composante du vecteur
  // d'état (dans un cadre multi-objets, le id-ème objet)
  void initializationComponent(int id) {
    // On utilise ici une marche aléatoire.
    Dynamics::randomWalk(this->state(id).p(),
                             GaussianNoise(0,3));
  }
  // Etape de prédiction.
  void predictionComponent(int id) {
  // On génère tout d'abord l'indice du modèle dynamique, ici de manière uniforme...
    this \rightarrow state (id).o().dynamic_model_ = (int)
       Distributions :: uniformSimulation (0,2);
    // ... propagation selon le modèle considéré : marche aléatoire ou modèle à vélocité
    // constante.
    if (this->state(id).o().dynamic_model_)
      Dynamics::randomWalk(this->state(id).p(),
                               GaussianNoise(0,3));
    else
      Dynamics:: constantVelocity(this->state(id).p(),
                                       GaussianNoise(0,3));
  }
  // Calcul de la vraisemblance.
  double likelihoodComponent(int id) {
    // Ici on utilise une comparaison entre les histogrammes de couleurs définis par
    // l'état getState(id) et par le modèle appris v_model_histo_.
    return Likelihood :: histogramCompare(ima_,
                                               this ->getState(id),
                                               (*v_model_histo_)[id]);
  }
  std :: vector <Histogram :: Histo*>* v_model_histo_;
  IplImage *ima_;
};
```

Enfin, il reste à définir la fonction main. L'initialisation de l'état peut se faire manuellement, ou bien, comme dans l'exemple proposé ci-après, grâce à un fichier d'initialisation, ou par un événement souris effectué par l'utilisateur sur la fenêtre de visualisation.

Fonction main

// STEP indique le pas utilisé dans chacune des dimensions de l'histogramme de // couleurs considéré.

```
const unsigned STEP = 25;
```

// Fonction main. Les arguments donnés en entrée du progamme sont -t video -f 100 // -m 1000 -o myInitFile.txt -n 200 -i "PathToMyVideo.avi". L'option -t gère le type de // flux (image, vidéo, ou webcam), -f indique la première image du flux (ici la 100-ième), // -m le nombre d'images maximal, -o le fichier d'initialisation, -n le nombre de // particules, et -i le chemin vers le flux d'entrée (ici la vidéo). int main(int argc, char** argv) // L'espace de noms IO gère automatiquement les arguments donnés à la fonction // main, et les redistribue aux composantes de la bibliothèque qui en ont l'usage. IO :: ReadArgs ra(argc, argv); // ReadImage permet de lire un flux vidéo, image, ou webcam. IO::ReadImage ri(&ra); // ReadInitFile récupère, s'il existe, le fichier d'initialisation du filtre particulaire. IO :: ReadInitFile <Point2d> rf(&ra); // DisplayImage permet d'afficher le résultat en temps réel. IO::DisplayImage di(argc, argv, "Example", 640, 480); // Nous définissons maintenant l'interface Utilisateur spécifiée précédemment... UserTest user; // ...ainsi que le filtre particulaire correspondant, qui considère ici des positions 2d, // une représentation par ellipse, et une classe de données auxiliaires OtherSpl. PF<UserTest, Position2d, Ellipse, OtherSpl> pf(&user, &ra); // Définition des histogrammes. std :: vector <Histogram :: Histo*> model_h; user.model_h_ = &model_h; int i = 0;typename State < Position2d, Ellipse, OtherSpl> TState;

```
// Lecture des arguments. La redistribution s'effectue également à ce moment.
ra . readArgs ();
// Initialisation de la graine aléatoire.
Distributions :: initialization ();
// La méthode suivante permet de garder trace de l'exécution du filtre particulaire.
```

// Ici, on enregistre la trajectoire, les poids et l'erreur, grâce au fichier de vérité
// terrain fournie.
pf.recordBegin("testTraj.txt",

```
"testWeights.txt",
"testError.txt",
```

```
"verite_terrain.txt");
// Tant que des images nous parviennent.
while (!ri.eof())
    // On récupère l'image courante.
    IplImage* input = ri.getImage();
    // obj_file récupère les données fournies par ReadInitFile.
    std :: vector < Point2d > obj_file = rf.getObjectAt(i);
    // obj_mouse récupère les interactions potentielles de l'utilisateur avec la
    // fenêtre de résultat.
    std :: vector < Point2d > obj_mouse = di.getData();
    // Si obj_file contient des informations au temps i, alors nous initialisons le ou
    // les nouveaux états, aux coordonnées désignées par obj_file, et les ajoutons à
    // l'estimation du filtre particulaire.
    for (unsigned k = 0; k < obj_file.size(); k++)
       ł
         TState s(Position2d(obj_file[k]), Ellipse(20,80),
                    OtherSpl());
         // Initialisation du modèle de référence.
         model_h.push_back(Histogram::initHisto(input,s,STEP));
         pf.addComponent(s);
       }
    // Si obj_mouse contient des informations au temps i, alors nous initialisons le
    // nouvel état, aux coordonnées et aux dimensions spécifiées par la sélection
    // avec la souris de l'utilisateur, et l'ajoutons à l'estimation du filtre particulaire.
    if (obj_mouse.size() == 2)
       {
         Point2d pos(obj_mouse[0].x_ + 0.5 * obj_mouse[1].x_{,}
                        obj_mouse[0].y_+ 0.5 * obj_mouse[1].y_);
         TState s(Position2d(pos), Ellipse(obj_mouse[1].x_,
                                                  obj_mouse[1].y_),
                    OtherSpl());
         // Initialisation du modèle de référence.
         model_h.push_back(Histogram::initHisto(input,s,STEP));
         pf.addComponent(s);
       }
    user.ima_ = input;
    // Algorithme principal, ici condensation. Celui-ci s'occupe, entre autres,
```

```
// d'appeler les fonctions définies dans l'interface utilisateur (initialisation une
// fois, puis prédiction et vraisemblance), pour toutes les particules.
```

```
pf.condensation();
```

// Affichage sur l'image des 10 particules de plus forts poids. Display2d :: fillHighestWeights (pf, input, cvScalar (255, 0, 255),

```
10, Display :: DisplayBoundary);
    // Affichage sur l'image de l'espérance de Monte-Carlo.
    Display2d :: fill (pf.getExpectedValue(), input,
                        cvScalar(255,255,255),
                        Display :: DisplayBoundary );
    // Affichage sur l'image des identifiants des objets.
    Display2d:: fillId (pf, input, cvScalar (255, 255, 255));
    // Affichage de l'image.
    di.display(input);
    ri.releaseImage();
    ++i;
  }
// Fin de l'enregistrement.
pf.recordEnd();
// Affichage de l'erreur quadratique moyenne.
std::cout << "RMSE:_" << pf.getRMSE() << std::endl;</pre>
// Fin de l'affichage.
di.end();
```

} }
Bibliographie

- B. D. O. Anderson et J. B. Moore. Optimal Filtering. Dover Publications, 2005.
- C. Andrieu, M. Davy, et A. Doucet. Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions. *IEEE Transactions on Signal Processing*, 51 (7):1762–1770, 2003a.
- C. Andrieu, N. de Freitas, A. Doucet, et M. I. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1-2):5–43, 2003b.
- E. Arnaud et E. Mémin. Partial Linear Gaussian Models for Tracking in Image Sequences Using Sequential Monte Carlo Methods. *International Journal of Computer Vision*, 74(1) : 75–102, 2007.
- E. Arnaud, E. Mémin, et B. Cernuschi-Frias. Conditional filters for image sequence-based tracking Application to point tracking. *IEEE Transactions on Image Processing*, 14(1) : 63–79, 2005.
- S. Arulampalam, S. Maskell, N. Gordon, et T. Clapp. A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2) :174–188, 2002.
- S. M. Arulampalam, B. Ristic, N. Gordon, et T. Mansell. Bearings-only tracking of manoeuvring targets using particle filters. *EURASIP Journal on Applied Signal Processing*, 15:2351–2365, 2004.
- J. Atif, C. Hudelot, G. Fouquier, I. Bloch, et E. Angelini. From generic knowledge to specific reasoning for medical image interpretation using graph-based representations. In *International Joint Conference on Artificial Intelligence*, pages 224–229, 2007.
- D. Avitzour. Stochastic simulation Bayesian approach to multitarget tracking. *IEEE Proceedings on Radar, Sonar and Navigation*, 142(2):41–44, 1995.
- Y. Bar-Shalom. Tracking and data association. Academic Press Professional, Inc., 1987.
- F. Bardet, T. Chateau, et D. Ramadasan. Suivi et Catégorisation Visuels en Temps Réel d'un Nombre Variable d'Objets : application au suivi de véhicules. In ORASIS'09 -Congrès des jeunes chercheurs en vision par ordinateur, 2009.

- F. Bardet, T. Chateau, et D. Ramadasan. Suivi et classification conjoints de multiples objets et de la source lumineuse par filtre particulaire MCMC. In *Reconnaissance des Formes et Intelligence Artificielle*, 2010.
- A. Ben-Hur, D. Horn, H. Siegelmann, et V. Vapnik. Support vector clustering. *The Journal* of Machine Learning Research, 2 :125–137, 2002.
- H. Benoudnine, S. Bartelmaos, et K. Abed-Meraim. An efficient IMM-UKF-Bias algorithm for mobile location in UMTS-FDD under NLOS conditions. In *IEEE International Symposium on Signal Processing and Information Technology*, pages 527–531, 2009.
- N. Bergman, A. Doucet, et N. Gordon. Optimal Estimation and Cramér-Rao Bounds for Partial Non-Gaussian State Space Models. *Annals of the Institute of Statistical Mathematics*, 53(1) :97–112, 2001.
- C. M. Bishop. Pattern Recognition and Machine Learning. Springer, 1 edition, 2007.
- A. Blake et M. Isard. Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion. Springer-Verlag, Secaucus, NJ, USA, 1998.
- A. Blake, M. Isard, et D. Reynard. Learning to track the visual motion of contours. Artificial Intelligence, 78(1-2):179 212, 1995.
- J. Blanco, J. González, et J. Fernández-Madrigal. An optimal filtering algorithm for nonparametric observation models in robot localization. In *IEEE International Conference on Robotics and Automation*, pages 461–466, 2008.
- I. Bloch. Fuzzy spatial relationships for image processing and interpretation: a review. *Image and Vision Computing*, 23(2):89–110, 2005.
- I. Bloch. Information Combination Operators for Data Fusion: A Comparative Review with Classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 26(1) : 52–67, 1996.
- Y. Boers et J. Driessen. Interacting multiple model particle filter. *IEEE Proceedings on Radar, Sonar and Navigation*, 150(5):344–349, 2003.
- P. Brasnett, L. Mihaylova, D. Bull, et N. Canagarajah. Sequential Monte Carlo tracking by fusing multiple cues in video sequences. *Image Vision Computing*, 25(8) :1217–1227, 2007.
- M. Bray, E. Koller-Meier, et L. V. Gool. Smart particle filtering for high-dimensional tracking. *Computer Vision and Image Understanding*, 106(1):116 – 129, 2007.
- D. Butnariu et E. P. Klement. *Triangular Norm-Based Measures and Games with Fuzzy Coalitions*. Kluwer Academic, 1993.
- Z. Cai et Z. Lin. Fuzzy Particle Filter Used for Tracking of Leukocytes. In *International Symposium on Intelligent Information Technology Application Workshops*, pages 562–565, 2008.

- F. Caron, M. Davy, E. Duflos, et P. Vanheeghe. Fusion de capteurs potentiellement défaillants par filtrage particulaire. In *Colloque sur le traitement du signal et des images* (*GRETSI*), pages 361–364, 2005.
- F. Caron, M. Davy, E. Duflos, et P. 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.
- J. Carpenter, P. Clifford, et P. Fearnhead. Improved particle filter for nonlinear problems. *IEEE Proceedings on Radar, Sonar and Navigation*, 146(1):2–7, 1999.
- G. Casella, J. Ferrándiz, D. Peña, D. Insua, J. Bernardo, P. García-López, A. González, J. Berger, A. Dawid, T. Diciccio, M. Wells, P. Gustafson, L. Wasserman, E. George, J. Liu, X.-L. Meng, A. Philippe, J. Schafer, et R. Strawderman. Statistical inference and Monte Carlo algorithms. *TEST: An Official Journal of the Spanish Society of Statistics and Operations Research*, 5(2) :249–344, 1996.
- W.-Y. Chang, C.-S. Chen, et Y.-D. Jian. Visual Tracking in High-Dimensional State Space by Appearance-Guided Particle Filtering. *IEEE Transactions on Image Processing*, 17(7) : 1154–1167, 2008.
- P.-H. Chen, R.-E. Fan, et C.-J. Lin. A study on SMO-type decomposition methods for support vector machines. *IEEE Transactions on Neural Networks*, 17(4):893–908, 2006.
- Z. Chen. Bayesian filtering: From Kalman filters to particle filters, and beyond. Technical report, McMaster University, 2003.
- S. Chib et E. Greenberg. Bayes inference in regression models with ARMA (p, q) errors. *Journal of Econometrics*, 64(1-2) :183–206, 1994.
- S. Chib et E. Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327–335, 1995.
- N. Chopin. Central Limit Theorem for Sequential Monte Carlo Methods and Its Application to Bayesian Inference. *The Annals of Statistics*, 32(6) :2385–2411, 2004.
- R. Collins, Y. Liu, et M. Leordeanu. On-Line Selection of Discriminative Tracking Features. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(1):1631–1643, 2005.
- O. Colliot. *Représentation, évaluation et utilisation de relations spatiales pour l'interprétation d'images. Application à la reconnaissance de structures anatomiques en imagerie médicale.* PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2003.
- O. Colliot, O. Camara, et I. Bloch. Integration of Fuzzy Spatial Relations in Deformable Models - Application to Brain MRI Segmentation. *Pattern Recognition*, 39(8):1401–1414, 2006.
- C. Cortes et V. Vapnik. Support-Vector Networks. Machine Learning, 20(3):273–297, 1995.
- J. Cui, H. Zha, H. Zhao, et R. Shibasaki. Multi-modal tracking of people using laser scanners and video camera. *Image and Vision Computing*, 26(2):240–252, 2008.

- P. Del Moral. A uniform convergence theorem for the numerical solving of the nonlinear filtering problem. *Journal of Applied Probability*, 35(4):873–884, 1998.
- P. Del Moral. *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer Verlag, 2004.
- P. Del Moral et A. Guionnet. Central limit theorem for nonlinear filtering and interacting particle systems. *Annals of Applied Probability*, 9(2):275–297, 1999.
- P. Del Moral et A. Guionnet. Large deviations for interacting particle systems: applications to non-linear filtering. *Stochastic processes and their applications*, 78(1):69–95, 1998.
- P. Del Moral, A. Doucet, et A. Jasra. On adaptive resampling procedures for sequential Monte Carlo methods. Technical report, INRIA, 2008.
- S. Derrode et W. Pieczynski. Signal and image segmentation using pairwise Markov chains. *IEEE Transactions on Signal Processing*, 52(9) :2477–2489, 2004.
- A. Deruyver et Y. Hodé. Constraint satisfaction problem with bilevel constraint: application to interpretation of over-segmented images. *Artificial intelligence*, 93(1-2):321–335, 1997.
- A. Deruyver, Y. Hodé, E. Leammer, et J. Jolion. Adaptive Pyramid and Semantic Graph: Knowledge Driven Segmentation. In *Graph-based representations in pattern recognition*, pages 213–222, 2005.
- F. Desbouvries et W. Pieczynski. Particle filtering with pairwise Markov processes. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 6, pages 705–708, 2003a.
- F. Desbouvries et W. Pieczynski. Particle filtering in pairwise and triplet Markov chains. In *Proceedings of the Nonlinear Signal and Image Processing Workshop*, pages 8–11, 2003b.
- I. Dhillon, Y. Guan, et B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *International Conference on Knowledge discovery and data mining*, pages 551–556, 2004.
- F. Dornaika, F. Davoine, et M. Dang. 3D head tracking by particle filters. In *International Workshop on Image Analysis for Multimedia Interactive Services*, 2004.
- R. Douc et E. Moulines. Limit theorems for weighted samples with applications to Sequential Monte Carlo Methods. *Annals of statistics*, 36(5) :2344–2376, 2008.
- A. Doucet et A. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In *Handbook of Nonlinear Filtering*. Oxford University Press. To appear, 2009.
- A. Doucet, N. De Freitas, K. Murphy, et S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Conference on Uncertainty in Artificial Intelligence*, pages 176–183, 2000a.
- A. Doucet, S. Godsill, et C. Andrieu. On Sequential Monte Carlo Sampling Methods for Bayesian Filtering. *Statistics and Computing*, 10:197–208, 2000b.
- A. Doucet, N. De Freitas, et N. Gordon, editors. *Sequential Monte Carlo methods in practice*. Springer, 2001a.

- A. Doucet, N. Gordon, et V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3) :613–624, 2001b.
- A. Doucet, B. Vo, C. Andrieu, et M. Davy. Particle Filtering for Multi-target Tracking and Sensor Management. In *Fifth International Conference on Information Fusion*, volume 1, pages 474–481, 2002.
- A. Doucet, M. Briers, et S. Senecal. Efficient block sampling strategies for sequential Monte Carlo methods. *Journal of Computational and Graphical Statistics*, 15(3):693–711, 2006.
- H. Driessen et Y. Boers. Efficient particle filter for jump Markov nonlinear systems. *IEE Proceedings of Radar, Sonar and Navigation*, 152(5) :323, 2005.
- W. Du et J. Piater. Tracking by Cluster Analysis of Feature Points and Multiple Particle Filters. In *IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 165– 170, 2005.
- Z. Duan, Z. Cai, et J. Yu. Fuzzy Adaptive Particle Filter Algorithm for Mobile Robot Fault Diagnosis. In *Neural Information Processing*, volume 4234, pages 711–720, 2006.
- D. Dubois et H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, Inc., 1980.
- S. Dubuisson. The computation of the Bhattacharyya distance between histograms without histograms. In *IEEE Image Processing Theory, Tools and Applications,* 2010.
- S. Duffner, J. Odobez, et E. Ricci. Dynamic Partitioned Sampling For Tracking With Discriminative Features. In *British Maschine Vision Conference*, 2009.
- A. El Abed. *Suivi multi-objets par filtrage particulaire dans un contexte de données incomplètes et/ou manquantes.* PhD thesis, Laboratoire d'informatique de Paris 6, 2008.
- A. El Abed, S. Dubuisson, et D. Bereziat. Enmim: Energetic normalized mutual information model for online multiple object tracking with unlearned motions. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 955–967, 2007.
- E. Erdem, S. Dubuisson, et I. Bloch. Particle Filter-Based Visual Tracking by Fusing Multiple Cues with Context-Sensitive Reliabilities. Technical Report 2010D002, Télécom ParisTech, 2010.
- R.-E. Fan, P.-H. Chen, et C.-J. Lin. Working Set Selection Using Second Order Information for Training Support Vector Machines. *Journal of Machine Learning Research*, 6 :1889– 1918, 2005.
- F. Faux et F. Luthon. Robust face tracking using colour Dempster-Shafer fusion and particle filter. In *International Conference on Information Fusion*, pages 1–7, 2006.
- D. Fink. A Compendium of Conjugate Priors. Technical report, Environmental Statistics. Department of Biology, Montana State University, 1997.
- F. Fleuret, J. Berclaz, R. Lengagne, et P. Fua. Multi-Camera People Tracking with a Probabilistic Occupancy Map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2008.

- C. Florin, N. Paragios, et J. Williams. Particle Filters, a Quasi-Monte Carlo Solution for Segmentation of Coronaries. In *Medical Image Computing and Computer-Assisted Intervention*, pages 246–253, 2005.
- G. Fouquier, J. Atif, et I. Bloch. Local Reasoning in Fuzzy Attribute Graphs for Optimizing Sequential Segmentation. In *Graph Based Representations in Pattern Recognition*, volume LNCS 4538, pages 138–147, 2007.
- D. Fox. Adapting the sample size in particle filters through KLD-sampling. *International Journal of Robotics Research*, 22(12):985, 2003.
- J. Garcia, J. M. Molina, J. A. Besada, et J. I. Portillo Garcia. Fuzzy approach for data association in image tracking. *Mathware and Soft Computing*, 10:117–129, 2003.
- A. Gelb. Applied optimal estimation. The MIT press, 1974.
- S. Geman et D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6) : 721–741, 1984.
- W. Gilks et D. Spiegelhalter. Markov chain Monte Carlo in practice. Chapman & Hall/CRC, 1996.
- W. R. Gilks et C. Berzuini. Following a moving target-Monte Carlo inference for dynamic Bayesian models. *Journal Of The Royal Statistical Society Series B*, 63(1):127–146, 2001.
- N. Gordon. A hybrid bootstrap filter for target tracking in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1):353–358, 1997.
- N. J. Gordon, D. J. Salmond, et A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings on Radar and Signal Processing*, 140(2):107–113, 1993.
- M. Grabisch et M. Sugeno. Fuzzy integrals and dual measures-application to pattern classification. In *Sino-Japan Joint Meeting on Fuzzy Sets and Systems*, 1990.
- M. Grabisch, T. Murofushi, et M. Sugeno. Fuzzy measure of fuzzy events defined by fuzzy integrals. *Fuzzy Sets and Systems*, 50(3):293 313, 1992.
- G. Grisetti, C. Stachniss, et W. Burgard. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.
- R. Guo, Z. Qin, X. Li, et J. Chen. Interacting Multiple Model Particle-type Filtering Approaches to Ground Target Tracking. *Journal of Computers*, 3(7):23, 2008.
- J. Handschin. Monte Carlo techniques for prediction and filtering of non-linear stochastic processes. *Automatica*, 6(4):555–563, 1970.
- W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
- F. Hausdorff. Set theory. Chelsea Publishing Company, 1962.
- K. Hotta. Adaptive Weighting of Local Classifiers by Particle Filter. In International Conference on Pattern Recognition, volume 2, pages 610–613, 2006.

- Y. Huang et P. Djuric. A hybrid importance function for particle filtering. *IEEE Signal Processing Letters*, 11(3):404–406, 2004.
- C. Hudelot, J. Atif, et I. Bloch. Fuzzy spatial relation ontology for image interpretation. *Fuzzy Sets and Systems*, 159(15):1929–1951, 2008.
- C. Hue. Méthodes séquentielles de Monte-Carlo pour le filtrage non linéaire multi-objets dans un environnement bruité. Applications au pistage multi-cibles et à la trajectographie d'entités dans des séquences d'images 2D. PhD thesis, Université de Rennes I, mention informatique, 2003.
- C. Hue, J.-P. Le Cadre, et P. Pérez. Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Transactions on Signal Processing*, 50(2):309–325, 2002.
- M. Hürzeler et H. R. Künsch. Monte Carlo approximations for general state-space models. *Journal of Computational and Graphical Statistics*, 7(2):175–193, 1998.
- N. Ikoma et S. Godsill. Extended object tracking with unknown association, missing observations, and clutter using particle filters. In *IEEE Workshop on Statistical Signal Processing*, pages 502–505, 2003.
- M. Isard. Visual Motion Analysis by Probabilistic Propagation of Conditional Density. PhD thesis, Oxford University, 1998.
- M. Isard et A. Blake. Condensation conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998a.
- M. Isard et A. Blake. A Mixed-State Condensation Tracker with Automatic Model-Switching. In *International Conference on Computer Vision*, pages 94–101, 1998b.
- M. Isard et J. MacCormick. BraMBLe: A Bayesian Multiple-Blob Tracker. In *International Conference on Computer Vision*, volume 2, pages 34–41, 2001.
- S. J. Julier et J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls,* volume 3, page 26, 1997.
- T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, 1967.
- R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions* of the ASME Journal of Basic Engineering, 82(D) :35–45, 1960.
- H. Kamel et W. Badawy. Fuzzy logic based particle filter for tracking a maneuverable target. In *48th Midwest Symposium on Circuits and Systems*, volume 2, pages 1537–1540, 2005.
- R. Karlsson et N. Bergman. Auxiliary particle filters for tracking a maneuvering target. In *IEEE Conference on Decision and Control*, volume 4, pages 3891–3895, 2000.
- S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, et K. R. K. Murthy. Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computing*, 13(3):637–649, 2001.

- Z. Khan, T. Balch, et F. Dellaert. MCMC-Based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 27(11):1805–1918, 2005.
- Y.-J. Kim, C.-H. Won, J.-M. Park, D. Hong, et M.-T. Lim. Fuzzy adaptive particle filter for a positioning beacon system. In *International Conference on Control, Automation and Systems*, pages 1198–1202, 2007.
- T. Kirubarajan, Y. Bar-Sralom, et D. Lerro. Bearings-only tracking of maneuvering targets using a batch-recursive estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 37(3):770–780, 2001.
- G. Kitagawa. Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.
- J. Klein, C. Lecomte, et P. Miche. Fast Color-Texture Discrimination: Application to Car Tracking. In *IEEE Intelligent Transportation Systems Conference*, pages 546–551, 2007.
- E. Klement et S. Weber. Fundamentals of a generalized measure theory. In U. Höhle et S. Rodabaugh, editors, *Mathematics of Fuzzy Sets. Logic, Topology, and Measure Theory, The Handbook of Fuzzy Sets Series*, pages 633–651. Kluwer Academic Publishers, 1999.
- S. Knerr, L. Personnaz, G. Dreyfus, J. Fogelman, A. Agresti, M. Ajiz, A. Jennings, F. Alizadeh, F. Alizadeh, et J. Haeberly. Single-layer learning revisited: A stepwise procedure for building and training a neural network. *Optimization Methods and Software*, 1:23–34, 1990.
- A. Kong, J. S. Liu, et W. H. Wong. Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association*, 89(425) :278–288, 1994.
- S. Kramer et H. Sorenson. Recursive Bayesian estimation using piece-wise constant approximations. *Automatica*, 24(6):789–801, 1988.
- C. Kreucher, A. Hero, et K. Kastella. Multiple Model Particle Filtering for Multi-Target Tracking. In *The Twelfth Annual Workshop on Adaptive Sensor Array Processing*, 2004.
- C. Kreucher, K. Kastella, et A. Hero. Multitarget tracking using the joint multitarget probability density. *IEEE Transactions on Aerospace and Electronic Systems*, 41(4) :1396–1414, 2005.
- H. J. Kushner. Numerical methods for stochastic control problems in continuous time. *SIAM journal on control and optimization*, 28(5):999–1048, 1990.
- O. Lanz. An information theoretic rule for sample size adaptation in particle filtering. In *International Conference on Image Analysis and Processing*, pages 317–322, 2007.
- D. Lesage, E. Angelini, I. Bloch, et G. Funka-Lea. Medial-based Bayesian tracking for vascular segmentation: Application to coronary arteries in 3D CT angiography. In *IEEE International Symposium on Biomedical Imaging*, pages 268–271, 2008.
- C. Liangcheng et Y. Huizhong. A Learning Method of Support Vector Machine Based on Particle Filters. In *Fifth International Conference on Natural Computation*, volume 1, pages 431–435, 2009.

- J. S. Liu. Metropolized Independent Sampling with Comparisons to Rejection Sampling and Importance Sampling. *Statistics and Computing*, 6:113–119, 1996.
- J. S. Liu et R. Chen. Blind Deconvolution Via Sequential Imputations. *Journal of the American Statistical Association*, 90(430):567–576, 1995.
- J. S. Liu et R. Chen. Sequential Monte Carlo Methods for Dynamic Systems. *Journal of the American Statistical Association*, 93 :1032–1044, 1998.
- S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982.
- K. Loquin et O. Strauss. Histogram density estimators based upon a fuzzy partition. *Statistics and Probability Letters*, 78(13):1863 1868, 2008.
- J. MacCormick. *Probabilistic modelling and stochastic algorithms for visual localisation and tracking*. PhD thesis, University of Oxford, 1999.
- J. MacCormick et A. Blake. Spatial Dependence in the Observation of Visual Contours. In *European Conference on Computer Vision*, volume 2, pages 765–781, 1998.
- J. MacCormick et A. Blake. A Probabilistic Exclusion Principle for Tracking Multiple Objects. *International Journal of Computer Vision*, 39(1):57–71, 2000.
- J. MacCormick et M. Isard. Partitioned Sampling, Articulated Objects, and Interface-Quality Hand Tracking. In European Conference on Computer Vision-Part II, pages 3–19, 2000.
- D. J. C. MacKay. Introduction to Monte Carlo methods. In *NATO Advanced Study Institute* on *Learning in graphical models*, pages 175–204, 1998.
- E. Maggio, F. Smeraldi, et A. Cavallaro. Combining colour and orientation for adaptive particle filter-based tracking. In *British Machine Vision Conference*, pages 659–668, 2005.
- S. McGinnity et G. Irwin. Multiple model bootstrap filter for maneuvering target tracking. *IEEE Transactions on Aerospace and Electronic Systems*, 36(3):1006–1012, 2000.
- N. Metropolis et S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247) :335–341, 1949.
- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, et E. Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6) : 1087–1092, 1953.
- S. P. Meyn et R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, London, 1993.
- F. Moreno-Noguer, A. Sanfeliu, et D. Samaras. Dependent multiple cue integration for robust tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4): 670–685, 2008.
- R. Muñoz-Salinas, E. Aguirre, M. García-Silvente, et A. Gonzalez. A multiple object tracking approach that combines colour and depth information using a confidence measure. *Pattern Recognition Letters*, 29(10) :1504–1514, 2008a.

- R. Muñoz-Salinas, M. García-Silvente, et R. M. Carnicer. Adaptive multi-modal stereo people tracking without background modelling. *Journal of Visual Communication and Image Representation*, 19(2):75–91, 2008b.
- R. Muñoz-Salinas, R. Medina-Carnicer, F. Madrid-Cuevas, et A. Carmona-Poyato. People detection and tracking with multiple stereo cameras using particle filters. *Journal of Visual Communication and Image Representation*, 20(5):339 – 350, 2009.
- M. Navara. Probability theory of fuzzy events. In *Fourth Conference of the European Society for Fuzzy Logic and Technology*, pages 325–329, 2005.
- O. Nempont. Modèles structurels flous et propagation de contraintes pour la segmentation et la reconnaissance d'objets dans les images. Application aux structures normales et pathologiques du cerveau en IRM. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 2009.
- O. Nempont, J. Atif, E. Angelini, et I. Bloch. Structure segmentation and recognition in images guided by structural constraint propagation. In *European Conference on Artificial Intelligence*, pages 621–625, 2008.
- K. Nummiaro, E. Koller-Meier, et L. V. Gool. Object Tracking with an Adaptive Color-Based Particle Filter. In *Symposium for Pattern Recognition of the DAGM*, pages 353–360, 2002.
- K. Okuma, A. Taleghani, N. de Freitas, J. Little, et D. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*, volume 3021, pages 28–39, 2004.
- M. Orton et W. Fitzgerald. A Bayesian approach to tracking multiple targets using sensor arrays and particle filters. *IEEE Transactions on Signal Processing*, 50(2):216–223, 2002.
- I. Patras et E. Hancock. Coupled Prediction Classification for Robust Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9) :1553–1567, 2010.
- V. Pavlovic, J. M. Rehg, et J. Maccormick. Learning Switching Linear Models of Human Motion. In *Neural Information Processing Systems*, pages 981–987, 2000.
- P. Pérez et J. Vermaak. Bayesian tracking with auxiliary discrete processes. application to detection and tracking of objects with occlusions. In *International Conference on Computer Vision - Workshop on Dynamical Vision*, pages 190–202, 2005.
- P. Pérez, A. Blake, et M. Gangnet. Jetstream: Probabilistic Contour Extraction with Particles. In *International Conference on Computer Vision*, pages 524–531, 2001.
- P. Pérez, C. Hue, J. Vermaak, et M. Gangnet. Color-Based Probabilistic Tracking. In *European Conference on Computer Vision-Part I*, pages 661–675, 2002.
- P. Pérez, J. Vermaak, et A. Blake. Data fusion for visual tracking with particles. *Proceedings* of the IEEE, 92(3):495–513, 2004.
- I. Perfilieva. Fuzzy transforms: Theory and applications. *Fuzzy Sets and Systems*, 157(8) : 993–1023, 2006.

- R. Péteri et O. Šiler. Object tracking using joint-visible and thermal-infrared video sequences. Technical report, Université de La Rochelle. Laboratoire Mathématiques, Image et Applications, 2009.
- B. Pham. Representation of Fuzzy Shapes. In *International Workshop on Visual Form*, pages 239–248, 2001.
- B. Pham et J. Zhang. A fuzzy shape specification system to support design for aesthetics. In L. Reznik et V. Kreinovich, editors, *Soft Computing in Measurement and Information*, pages 204–252. Springer, 2001.
- D. Pham. Fuzzy clustering with spatial constraints. In *IEEE International Conference on Image Processing*, pages 65–68, 2002.
- M. Pitt et N. Shephard. Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.
- J. C. Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*, pages 61–74, 2000.
- F. Porikli. Integral histogram: a fast way to extract histograms in Cartesian spaces. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 829–836, 2005.
- Y. Rathi, N. Vaswani, et A. Tannenbaum. A Generic Framework for Tracking Using Particle Filter With Dynamic Shape Prior. *IEEE Transactions on Image Processing*, 16(5):1370 –1382, 2007.
- B. Riečan et D. Mundici. Probability on MV-algebras. In *Handbook of measure Theory*, chapter 21, pages 869–909. Elsevier, 2002.
- C. P. Robert et G. Casella. Monte Carlo Statistical Methods. Springer-Verlag, 2005.
- H. Ryu et M. Huber. A particle filter approach for multi-target tracking. In *International Conference on Intelligent Robots and Systems,* pages 2753–2760, 2007.
- S. Särkkä, A. Vehtari, et J. Lampinen. Rao-Blackwellized Monte Carlo data association for multiple target tracking. In *Seventh International Conference on Information Fusion*, volume 7, pages 583–590, 2004.
- B. Schölkopf, C. Burges, et V. Vapnik. Extracting support data for a given task. In *First International Conference on Knowledge Discovery & Data Mining*, pages 252–257, 1995.
- D. Schulz, W. Burgard, D. Fox, et A. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1665–1670, 2001.
- L. Simon et J.-P. Tarel. Utilisation de la distance de Hausdorff dans un SVM? In *Manifestation des Jeunes Chercheurs STIC (Majecstic'06)*, Lorient, France, 2006.
- K. Smith et D. Gatica-Perez. Order Matters: A Distributed Sampling Method for Multi-Object Tracking. In *British Maschine Vision Conference*, pages 25–32, 2004.

- K. Smith, D. Gatica-Perez, et J.-M. Odobez. Using Particles to Track Varying Numbers of Objects. In *Computer Vision and Pattern Recognition*, volume 1, pages 962–969, 2005.
- K. C. Smith. *Bayesian methods for visual multi-object tracking with applications to human activity recognition.* PhD thesis, École Polytechnique Fédérale de Lausanne, 2007.
- C. Snyder, T. Bengtsson, P. Bickel, et J. Anderson. Obstacles to High-Dimensional Particle Filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.
- A. Soto. Self Adaptive Particle Filter. In International Joint Conference on Artificial Intelligence, pages 1398–1403, 2005.
- O. Straka et M. Šimandl. Sample size adaptation for particle filters. In *Symposium on Automatic control in aerospace*, pages 437–442, 2005.
- O. Straka et M. Simandl. Particle filter adaptation based on efficient sample size. In *Symposium on System Identification*, pages 991–996, 2006.
- M. Sugeno. *Theory of fuzzy integrals and its applications*. PhD thesis, Tokyo Institute of Technology, 1974.
- A. Tanács, C. Domokos, N. Sladoje, J. Lindblad, et Z. Kato. Recovering Affine Deformations of Fuzzy Shapes. In *Scandinavian Conference on Image Analysis*, volume 5575, pages 735–744, 2009.
- H. Tanizaki et R. S. Mariano. Nonlinear Filters Based On Taylor Series Expansions. *Communications in Statistics Theory and Methods*, 25:1261–1282, 1996.
- N. Thacker, F. Aherne, et P. Rockett. The Bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 34(4):363–368, 1997.
- L. Tierney. Markov chains for exploring posterior distributions. *Annals of Statistics*, 22 : 1701–1762, 1994.
- J. Triesch et C. Malsburg. Democratic integration: Self-organized integration of adaptive cues. *Neural computation*, 13(9):2049–2074, 2001.
- M. Vanegas, I. Bloch, et J. Inglada. Fuzzy Spatial Relations for High Resolution Remote Sensing Image Analysis: The Case of "To Go Across". In *IEEE International Geoscience and Remote Sensing Symposium*, volume 4, pages 773–776, 2009.
- M.-C. Vanegas, I. Bloch, et J. Inglada. Searching aligned groups of objects with fuzzy criteria. In *International Conference on Information Processing and management of Uncertainty in Knowledge-Based Systems, IPMU*, pages 605–613, 2010.
- N. Vaswani. Particle Filtering for Large-Dimensional State Spaces With Multimodal Observation Likelihoods. *IEEE Transactions on Signal Processing*, 56(10):4583–4597, 2008.
- J. Vermaak, M. Gangnet, A. Blake, et P. Pérez. Sequential Monte Carlo fusion of sound and vision for speaker tracking. In *International Conference on Computer Vision*, volume 1, pages 741–746, 2001.
- J. Vermaak, S. Godsill, et P. Pérez. Monte Carlo filtering for multi-target tracking and data association. *IEEE Transactions on Aerospace and Electronic Systems*, 41:309–332, 2005a.

- J. Vermaak, S. Maskell, M. Briers, et P. Pérez. Bayesian visual tracking with existence process. In *IEEE International Conference on Image Processing*, volume 1, pages 721–725, 2005b.
- J. Wang, D. Zhao, W. Gao, et S. Shan. Interacting multiple model particle filter to adaptive visual tracking. In *International Conference on Image and Graphics*, pages 568–571, 2004.
- J. Wang, X. Chen, et W. Gao. Online selecting discriminative tracking features using particle filter. In *IEEE Computer Society Conference on Computer Vision and Pattern Recog-nition*, volume 2, pages 1037–1042, 2005.
- Y. Wang, J. Wu, et A. Kassim. Adaptive particle filter for data fusion of multiple cameras. *The Journal of VLSI Signal Processing*, 49(3):363–376, 2007.
- B. L. Welch. The generalization of "Student's" problem when several different population variances are involved. *Biometrika*, 34(1-2):28–35, 1947.
- G. Welch et G. Bishop. An Introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill, 1995.
- H. Wu, F. Sun, et H. Liu. Fuzzy Particle Filtering for Uncertain Systems. *IEEE Transactions* on *Fuzzy Systems*, 16(5) :1114–1129, 2008.
- T.-F. Wu, C.-J. Lin, et R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004.
- X. Xu et B. Li. Rao-Blackwellised particle filter for tracking with application in visual surveillance. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 17–24, 2005.
- R. Yager. Uncertainty representation using fuzzy measures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* 32(1) :13–20, 2002.
- R. R. Yager. On a general class of fuzzy connectives. *Fuzzy Sets and Systems*, 4(3):235 242, 1980.
- R. R. Yager. Connectives and quantifiers in fuzzy sets. *Fuzzy Sets and Systems*, 40(1):39 75, 1991.
- M. Yang, T. Yu, et Y. Wu. Game-Theoretic Multiple Target Tracking. In *International Conference on Computer Vision*, 2007.
- W. Ye, H. Liu, F. Sun, et M. Gao. Vehicle tracking based on co-learning particle filter. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2979–2984, 2009.
- L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning I. *Information Sciences*, 8(3):199–249, 1975.
- L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1): 3–28, 1978.
- L. A. Zadeh. Probability Measures of Fuzzy Events. *Journal of Mathematical Analysis and Applications*, 23(2):421–427, 1968.

- G. Zhang, Z. Yuan, N. Zheng, X. Sheng, et T. Liu. Visual Saliency Based Object Tracking. *The Tenth Asian Conference on Computer Vision*, pages 193–203, 2010.
- D. Zotkin, R. Duraiswami, et L. Davis. Joint audio-visual tracking using particle filters. *EURASIP Journal on Applied Signal Processing*, 2002(1):1164, 2002.