

N° d'ordre : 41129

MINES DOUAI



UNIVERSITE DE LILLE 1



THESE

présentée en vue
d'obtenir le grade

DOCTEUR

en

Spécialité : Sciences de l'Informatique

par

Imen SAFFAR

**DOCTORAT DELIVRE CONJOINTEMENT
PAR MINES DOUAI ET L'UNIVERSITE DE LILLE 1**

Titre de la thèse :

Vers une agentification de comportements observés :
une approche originale basée sur l'apprentissage
automatique pour la simulation d'un environnement réel.

Soutenue le 19 juin 2013 devant le jury d'examen :

Président	René MANDIAU, Professeur, UVHC
Rapporteur	Abdelhamid BOUCHACHIA, Professeur associé, Université de Bournemouth
Rapporteur	Stéphane ESPIE, Directeur de recherche, IFSTTAR
Examineur	Laurent VERCOUTER, Professeur, INSA de Rouen
Directeur de thèse	Stéphane LECOEUCE, Professeur, Mines Douai
Encadrant	Arnaud DONIEC, Maître assistant, Mines Douai

Laboratoire d'accueil : Unité de Recherche Informatique et Automatique de Mines Douai

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

Thèse cofinancée par la région Nord-Pas-de-Calais



Remerciements

Je ne pourrais mieux qualifier mes quatre ans de thèse que de moments forts en émotion ! Je suis passée par des phases difficiles suite à la perte d'une personne des plus chères à mes yeux qui n'est d'autre que mon défunt père.

Je suis également passée par l'une des périodes les plus riches de ma vie, aussi bien du côté scientifique que relationnel : j'ai eu la chance d'être entourée de chercheurs, venant des quatre coins du monde, qui ont été une véritable source d'inspiration, de soutien et d'amitié indéfectible.

Je tiens à remercier en premier lieu Stéphane Lecoeuche, mon directeur de thèse, ainsi que Arnaud Doniec, mon encadrant, pour leur confiance, leurs précieux conseils et leur patience.

J'aimerais exprimer toute ma gratitude à toutes les personnes qui, à un moment donné, ont participé à la mise en œuvre de mon projet de thèse ou, tout simplement, à des réflexions philosophiques en particulier Anthony Fleury, Laurent Bako, Vincent Perradin et Jacques Boonaert. Je n'oublierai jamais nos débats interminables, pendant les pauses café, sur la possessivité, les acquis, la vie ou l'haut-delà.

Ce travail doctoral n'aurait jamais abouti sans le soutien de ma famille, en particulier de mon père dans un premier temps et de ma mère tout le long. Je n'ai pas de mots assez forts qui permettraient d'exprimer ma reconnaissance et mon amour à leur égard.

Pour finir, je dédie cette thèse à ma mère et à la mémoire de mon père.

Table des matières

Introduction générale	1
1 Méthodologies de conception agent	5
1.1 Introduction	5
1.2 Simulations des espaces de vente : Analyse du comportement du consommateur	7
1.2.1 Outils de simulation basés sur des comportements abstraits	7
1.2.2 Outils de simulation basés sur des comportements de clients réels	9
1.3 Les systèmes multi-agents (SMA) et simulation	11
1.3.1 L'agent et ses caractéristiques	12
1.3.2 Les types d'agents	12
1.3.3 Les SMA et leurs caractéristiques	13
1.3.4 Les SMA réactifs	13
1.3.5 Les simulations multi-agents	14
1.3.6 L'émergence dans les simulations multi-agents	14
1.4 Méthodes de conception de simulation multi-agents	15
1.4.1 Démarche au niveau micro	16
1.4.1.1 Approche par méthodes de conception	16
1.4.1.2 Approche par ajustement des paramètres	16
1.4.2 Démarche au niveau macro	17
1.4.3 Conception de simulation participative	18
1.4.4 Conception de simulation basée sur les interactions	18
1.5 Combinaison du Data Mining et des simulations agents	20
1.5.1 Application exogène	22
1.5.2 Application endogène	23
1.6 Conclusion	23
2 Extraction de connaissances à partir de l'observation	25
2.1 Introduction	25
2.2 Présentation des besoins liés à l'observation du monde réel	26
2.3 Etat de l'art sur les techniques en traitement de trajectoires	28
2.4 Extraction de comportements par apprentissage de trajectoires	29
2.4.1 Réseaux Bayésiens Dynamiques Récurrents	30
2.4.2 Le Modèle de Markov Caché	32
2.4.3 Modèles dynamiques commutés	38
2.4.4 Modèle de mélange pour la classification automatique	41

2.4.4.1	Le maximum de vraisemblance	42
2.4.4.2	L'algorithme EM	43
2.4.4.3	EM pour les mélanges de modèles de régression	45
2.5	Discussion et conclusion	48
3	Méthodologies et contribution	49
3.1	Introduction	49
3.2	Méthodologie de conception de simulation multi-agents	51
3.2.1	Prérequis	54
3.2.2	Comportements individuels des agents	55
3.2.2.1	Discrétisation et annotation de l'environnement observé et simulé	56
3.2.2.2	Motivations des agents à partir de l'environnement observé	57
3.2.3	Données collectives observées pour la vérification du modèle	61
3.3	Extraction de comportements à partir d'observations	63
3.3.1	Application de l'algorithme EM	63
3.3.1.1	Détermination du nombre optimal de clusters	64
3.3.1.2	Trajectoires moyennes : comportements prototypes	65
3.3.1.3	Vérification par classification ascendante hiérarchique	69
3.3.1.4	Traduction des trajectoires en motivations	71
3.3.2	Application des MMC	73
3.3.2.1	Le modèle de la structure générique	73
3.3.2.2	Choix du nombre d'états cachés optimal du MMC	76
3.4	Discussion et conclusion	79
4	Implémentation et résultats	81
4.1	Introduction	81
4.2	Implémentation	82
4.2.1	Plateformes de simulation multi-agents	82
4.2.1.1	Etat de l'art des outils existants	82
4.2.1.2	Utilisation de Repast Symphony	84
4.2.1.3	Architecture de la simulation	85
4.2.2	Description du workflow lié à notre méthodologie	87
4.2.2.1	Vue globale	88
4.3	Résultats	92
4.3.1	Résultats avec EM	92
4.3.2	Résultats avec MMC	94
4.3.3	Synthèse	97
4.4	Transposition au commerce	98
4.5	Conclusion	102
	Conclusion et perspectives	105
A	Proposition d'automatisation du découpage de l'environnement	119

Table des figures

1.1	Méthode de codage du chromosome	9
1.2	Processus de modélisation de simulation multi-agents	15
1.3	Processus de simulation agent dans un contexte participatif [Guyot et Honiden, 2006]	19
1.4	Application du DM pour la vérification et la validation des simulations multi-agents [Baqueiro <i>et al.</i> , 2009]	22
2.1	Illustration du suivi de personnes dans un supermarché	27
2.2	Réseau Bayésien Récurrent pour la reconnaissance de niveau d'agitation globale d'un groupe de personnes [Moënné-Loccoz <i>et al.</i> , 2003]	31
2.3	Traitement de trajectoires [Mlich et Chmelar, 2008]	34
2.4	Modélisation de trajectoires par MMC	35
2.5	Représentation en réseau bayésien dynamique du modèle de comportement sur deux laps de temps [Nguyen et Venkatesh, 2005]	37
2.6	Mécanisme de modélisation de trajectoire basé sur les modèles dynamiques commutés	39
2.7	Architecture du processus incrémental indépendant basé sur les modèles dynamiques commutés [Nascimento <i>et al.</i> , 2005]	40
2.8	Exemple de résultats de la méthode de mélanges de modèles de régression, extrait de [Gaffney et Smyth, 1999]	47
3.1	Méthodologie de construction de simulation multi-agents	53
3.2	Environnement étudié	54
3.3	Exemple d'une trajectoire suivie par un individu sur les images de la vidéo	56
3.4	L'environnement réel	57
3.5	Grille de l'environnement de la simulation	58
3.6	Exemple de trajectoire passant par plusieurs RI	58
3.7	Exemple de trajectoire composée de points irréguliers en présentant des concentrations au niveau des deux extrémités	60
3.8	Carte des points chauds et des points froids du monde réel, réalisée à partir des 682 trajectoires extraites de la vidéo	62
3.9	Coefficients de corrélation entre les images et la première.	62
3.10	Principe de la Validation Croisée avec $K = 4$	64
3.11	Les 24 trajectoires moyennes (3 par image)	66
3.12	Exemple de trajectoires moyennes complètes, superposées sur l'ensemble des instances des clusters correspondant	67
3.13	Trajectoires du cluster 18	67

3.14	Trajectoires du cluster 12	68
3.15	Coupe transversale du dendrogramme entraînant 24 segments	70
3.16	Exemple d'instances	71
3.17	Exemple d'un MMC à 3 états	74
3.18	Etiquettes des régions de l'environnement du hall universitaire	75
3.19	Exemple d'une liste d'observations engendrée par un parcours de six états cachés . .	75
3.20	Dispersion de l'état	77
3.21	Diamètres de premier ordre en fonction du nombre d'états cachés	78
3.22	Zones correspondantes à quelques états cachés extraits du MMC à 20 états cachés .	78
3.23	Exemple de déplacement d'un agent dans la simulation suivant une liste de motivations correspondantes aux 5 observations obtenues par un MMC à 20 états	79
4.1	Modèle de développement sur Repast Symphony	85
4.2	Structure générale du modèle de la simulation	86
4.3	Workflow de la méthodologie retenue	87
4.4	Illustration d'un blob dans l'environnement étudié	88
4.5	L'interface de Repast Symphony pour la simulation du hall d'entrée du bâtiment . . .	91
4.6	Comparaison entre la carte de chaleur initiale et celle après l'emploi de EM	93
4.7	Comparaison entre la carte de chaleur initiale et celle après l'emploi de EM (Vue caméra)	94
4.8	Exemples de déplacements avec MMC à 8 états cachés	95
4.9	Comparaison avec carte de chaleur du MMC retenu	96
4.10	Cartes de chaleur des MMCs pour différents nombres d'états cachés	97
4.11	Comparaison entre la carte de chaleur initiale et celle après l'emploi du MMC à 20 états cachés (Vue caméra)	98
4.12	La méthodologie appliquée sur un espace de vente	100
4.13	Illustration de zones permutées (Board31/Board32 et Coffee1&2/Printer)	101
4.14	Cartes de chaleur obtenues, issues de EM, avant et après permutation des RI "Coffee1", "Coffee2" et "Printer" avec "Board31" et "Board32"	101
4.15	Classification hiérarchique d'individus à partir de plusieurs sources de données	108
4.16	Conjonction et classification d'individus à partir de plusieurs sources de données . . .	109
A.1	Le principe de Voronoï	120
A.2	Exemple de fusion avec 3 trajectoires	120
A.3	Partition de l'environnement utilisant le diagramme de Voronoï discret	121

Introduction générale

A l'époque actuelle, la consommation peut apparaître aux yeux de certains comme un moyen de satisfaire ses besoins et ses désirs. Ainsi, la production de biens, le concept qui est à l'origine de la consommation, ne se contente pas de répondre aux besoins primordiaux des consommateurs. Il permet d'assurer la pérennité des entreprises et le développement de l'économie en suscitant et renouvelant les besoins du consommateur. Ce phénomène est entretenu et stimulé par diverses prestations de services spécialisées dans le marketing.

Les professionnels du marketing ne se limitent plus à réaliser des analyses de marchés au niveau global pour mettre en place des opérations stratégiques ; ils s'intéressent de plus en plus à un niveau microscopique, au comportement individuel des consommateurs. Ils voient dans cette démarche la possibilité d'augmenter encore plus leurs performances économiques [Ladwein, 1999]. Cela impose que l'étude du comportement du consommateur soit abordée par plusieurs disciplines telles que l'économie, la psychologie, la sociologie et l'anthropologie ce qui a conduit à l'élaboration de nombreux modèles.

Divers types de commerce s'appuient sur la modélisation du comportement du consommateur, notamment dans les espaces de vente (supermarchés, magasins de vêtements, magasins d'électronique...). Dans l'hypothèse où l'on dispose d'un modèle représentatif des comportements des clients à l'intérieur de l'établissement, il serait possible de simuler l'environnement, les personnes qui y évoluent et leurs interactions. Ainsi, il est envisageable d'étudier et d'analyser "*in vitro*" les paramètres du modèle (tels que les motivations des consommateurs), l'influence de l'environnement (telle que la disposition des linéaires) ou tout autre élément pertinent dans le processus de vente.

Nous nous intéressons dans cette thèse, qui s'inscrit dans la cadre du Pôle de Compétitivité Industries du Commerce (Picom), à l'observation et à la simulation du comportement du consommateur dans un espace de vente. Nous proposons une méthodologie prospective applicable à la simulation d'un environnement de commerce et des comportements de consommateurs y évoluant dont l'objectif sera d'évaluer les influences de différentes stratégies commerciales telles que l'emplacement des différents linéaires ou des stands promotionnels.

Pour aboutir à un tel outil, la principale difficulté réside dans la capacité à modéliser un système complexe et non-stationnaire dans lequel siègent de nombreuses activités humaines. Vient s'ajouter à cela le fait que la représentation des comportements doit éviter toute forme de subjectivité afin de ne pas dénaturer le modèle. Dans cette optique, deux problématiques se posent : comment formalise-t-

on ce type de problème de manière aussi objective que possible ? Quel type de modèle doit-on mettre en place pour représenter ces comportements et tenir compte de leur diversité ?

Il existe deux approches différentes, citées dans [Parunak *et al.*, 1998], permettant de construire un modèle de simulation. La plus ancienne et couramment utilisée est celle qui porte sur la modélisation basée sur des équations. La deuxième porte sur la modélisation comportementale reposant sur le paradigme "agent". Le recours à la première approche est généralement motivé par la possibilité de modéliser le système par un ensemble d'équations qu'on cherche à évaluer. Quant à la seconde approche, le modèle est constitué d'un ensemble d'agents encapsulant des comportements multiples correspondant aux individus qui composent le système que l'on cherche à reproduire. Ainsi, explorer une modélisation basée sur le paradigme d'agent et l'identification de modèles de comportements est en adéquation avec notre problématique.

La première difficulté que nous pouvons rencontrer lors de la modélisation d'un système multi-agents est l'identification des éléments qui le composent. A-t-on intérêt à modéliser chaque individu observé par un type d'agent ? Cela paraît logique et anodin, dans le cas où on cherche à simuler un magasin de vente sur la base d'une journée d'observation. Cependant, qu'advient-il si nous considérons une base d'observation d'une année ? Il faudrait des milliers voir des millions de types d'agent créés dans la simulation (selon le directeur général d'Auchan France, par exemple, le plus petit des magasins Auchan accueille en moyenne 4000 clients par jour¹). Afin de contourner cette éventuelle complication, il est judicieux de déterminer les acteurs essentiels à modéliser. De cette manière, nous réduisons significativement le nombre de types d'agent à expérimenter sans pour autant amoindrir le réalisme de la simulation.

Dans l'hypothèse où les types d'agent essentiels à simuler sont déterminés, un autre verrou se pose. La notion de comportement d'un consommateur dans un espace de vente n'est pas prédéfinie à l'avance. De nombreux paramètres sont susceptibles de mettre en valeur et caractériser un comportement. Pour ne s'intéresser qu'aux comportements observables au sein du magasin, certains sont perceptibles et potentiellement pertinents à relever. Par exemple, il est possible de suivre les chemins suivis par chacun des consommateurs. De ces comportements, on est potentiellement capable de déterminer et extraire des caractéristiques comportementales utiles à la conception d'agents de type consommateur. Une fois ces comportements reproduits en simulation, d'autres informations pertinentes peuvent être estimées, notamment, l'attrait de certains produits et, de façon plus générale, des linéaires. Cela permet d'évaluer partiellement l'attractivité du point de vente.

Après avoir pris connaissance de ces problématiques, nous proposons de fonder intégralement notre méthodologie sur des modèles de comportements situés représentant le déplacement des individus. Ces comportements seront construits de façon automatique à partir de traitements d'apprentissage non supervisé afin d'éviter un paramétrage subjectif et une étude empirique fastidieuse spécifique au domaine d'application. La démarche sera ensuite d'exploiter les comportements de déplacement, au sein d'une simulation reproduisant le magasin cible, résultant de l'observation.

1. Article apparu sur le site <http://www.lsa-conso.fr/le-patron-d-auchan-france-devoile-7-solutions-anticrise-en-hyper,128012>

Enfin, restera à valider le résultat de la simulation, sachant que la conception d'un système multi-agents est plus intuitive que formelle. Il faut donc s'assurer de l'adéquation entre le modèle et le phénomène réel. En d'autres termes, le modèle doit être capable de révéler des informations identifiables sur le système modélisé. Celles-ci doivent être assimilables, par le biais d'une comparaison, à des informations observables provenant de données empiriques [Amblard *et al.*, 2006]. L'observation pourra alors jouer double rôle, un premier en tant que moyen de découverte des comportements, et un second en tant que moyen de validation.

Nous abordons, dans cette thèse, une méthodologie multi-agents faisant face à de multiples verrous, notamment :

- la représentation objective des comportements ;
- l'identification des types d'agent ;
- la définition des comportements situés ;
- l'adéquation entre le modèle et le phénomène réel.

De plus, elle touche à différents domaines de compétence, notamment l'apprentissage et la simulation multi-agents. Afin de présenter notre contribution, le manuscrit sera organisé en quatre chapitres.

Le chapitre 1 présente, en premier lieu, un état de l'art sur des méthodologies portant sur l'analyse des comportements des consommateurs par le biais de simulations. Nous abordons ensuite des notions propres aux systèmes multi-agents et simulations comportementales, évoquant les contraintes de conception et de vérification. Enfin, différentes méthodologies de conception de simulation multi-agents sont exposées mettant en avant leurs avantages et leurs inconvénients.

Le chapitre 2 se focalise sur des techniques, issues de la littérature, portant sur l'extraction de comportements de déplacements à partir d'observations. Ce qui permet de recueillir automatiquement des comportements fidèles à la réalité. Ces méthodes s'appuient sur des techniques d'apprentissage non supervisé. Nous nous intéressons particulièrement à deux d'entre elles, notamment la technique d'Estimation-Maximisation (EM) [McLachlan et Krishnan, 1996] et les Modèles de Markov Cachés (MMC) [Rabiner et Juang, 1986].

Le chapitre 3 présente notre contribution. Nous y exposons notre méthodologie générale pour la conception d'une simulation multi-agents basée sur l'observation du monde réel. Nous nous focalisons, en particulier, sur la partie extraction et agentification des comportements observés. Nous consacrons, de ce fait, deux sections principales traitant, d'une part, une agentification basée sur la technique de EM, et d'autre part, une agentification basée sur les MMC. Dans le but d'expérimenter notre proposition, nous appliquons notre méthodologie sur un environnement simple (dans la perspective de réaliser une transposition sur un magasin de vente).

Le chapitre 4 détaille les différentes étapes permettant d'appliquer notre méthodologie en mettant l'accent sur la partie simulation, l'outil utilisé et l'architecture développée. Les résultats de la simulation obtenus suite à l'insertion des modèles produits par EM d'une part et par MMC d'autre part sont présentés et évalués. Enfin, nous consacrons la dernière partie à la transposition à un magasin.

Pour finir, nous concluons ce manuscrit par une synthèse générale sur ce qui a été entrepris. Nous énumérons, par ailleurs, les perspectives d'études complémentaires à notre approche en les scindant en perspectives théoriques et perspectives applicatives.

Chapitre 1

Méthodologies de conception agent

Sommaire

1.1	Introduction	5
1.2	Simulations des espaces de vente : Analyse du comportement du consommateur	7
1.2.1	Outils de simulation basés sur des comportements abstraits	7
1.2.2	Outils de simulation basés sur des comportements de clients réels	9
1.3	Les systèmes multi-agents (SMA) et simulation	11
1.3.1	L'agent et ses caractéristiques	12
1.3.2	Les types d'agents	12
1.3.3	Les SMA et leurs caractéristiques	13
1.3.4	Les SMA réactifs	13
1.3.5	Les simulations multi-agents	14
1.3.6	L'émergence dans les simulations multi-agents	14
1.4	Méthodes de conception de simulation multi-agents	15
1.4.1	Démarche au niveau micro	16
1.4.2	Démarche au niveau macro	17
1.4.3	Conception de simulation participative	18
1.4.4	Conception de simulation basée sur les interactions	18
1.5	Combinaison du Data Mining et des simulations agents	20
1.5.1	Application exogène	22
1.5.2	Application endogène	23
1.6	Conclusion	23

1.1 Introduction

La grande diversité et la profusion de produits mis à disposition des consommateurs rend de plus en plus complexe la tâche d'aménagement dans des espaces de vente. En vue d'améliorer leur agencement et leur performance commerciale, il est nécessaire de connaître l'attractivité des produits, d'estimer ou mesurer l'affluence du magasin, d'étudier les comportements des consommateurs, et enfin de disposer des stratégies de vente que le gestionnaire pourrait mettre en application.

De telles analyses sont difficiles à réaliser *in vivo*, le recours à une simulation peut donc s'avérer être un véritable gain de temps et de performance. En effet, exercer une action sur un élément réel du magasin pour pouvoir observer ou mesurer la conséquence, n'est pas toujours possible. Disposant d'un modèle représentatif des comportements et des interactions qui se produisent dans un espace de vente, il est envisageable de simuler et de pronostiquer les influences des actions potentielles, notamment les différentes stratégies commerciales relatives à l'agencement de la surface de vente. Cette simulation fournirait ainsi un outil d'aide à la décision destiné au manager de l'espace de vente. Il permettrait de reproduire les interactions qui se produisent dans le magasin et d'analyser :

- l'influence des variables d'entrée, telles que l'affluence, la configuration de l'espace de vente ;
- les paramètres associés à l'espace de vente et aux interactions consommateurs-produits, tels que les motivations des agents consommateurs ou encore l'attractivité des produits ;
- les perturbations telles que l'influence de l'environnement.

En jouant sur les paramètres, un tel outil de simulation donne le moyen d'étudier *a priori* l'impact de l'agencement du magasin, les meilleurs emplacements des promotions, la localisation des services d'aide au consommateur voire tout autre élément pertinent aux processus de vente.

Cependant, modéliser un tel système dans le but de réaliser cet outil n'est pas une tâche aisée étant donné la complexité et le caractère dynamique des phénomènes à reproduire. Ici réside l'utilité de la simulation basée sur les systèmes multi-agents, généralement utilisée lorsque le comportement du système est trop complexe pour être décrit analytiquement. Néanmoins, la difficulté de construire un modèle de simulation multi-agents d'un espace de vente persiste dans le sens où il faut s'assurer de disposer d'une représentation réaliste, d'une adéquation suffisante et réaliste avec l'aménagement du magasin et de l'ensemble des interactions qui s'y produisent. Il faut donc d'une part une bonne modélisation des comportements des consommateurs, et d'autre part, une simulation réaliste basée sur le paradigme d'agent et des interactions associées.

Les systèmes multi-agents constituent un des champs d'étude de l'Intelligence Artificielle, à l'intersection du génie logiciel et de l'informatique distribuée. Apparus il y a plus de 25 ans, les systèmes multi-agents ont atteint aujourd'hui un niveau de maturité suffisamment important pour être mis en oeuvre dans différentes applications particulièrement complexes. De plus en plus d'entreprises trouvent ainsi cette approche attractive et ont tendance à l'adopter dans leurs outils commerciaux [Parunak, 2000].

Dans la première partie de ce chapitre, nous allons étudier certains travaux qui traitent des simulations des espaces de vente. L'objectif lié à cette partie est de situer notre problématique par rapport aux approches antérieures. La seconde partie est dédiée aux notions de base sur les simulations multi-agents en passant par les systèmes multi-agents et leurs composants. Enfin, les deux dernières parties portent sur une étude bibliographique des approches de conception de simulation multi-agents, qui permettra de situer notre méthodologie développée dans les chapitres suivants.

1.2 Simulations des espaces de vente : Analyse du comportement du consommateur

De nos jours, la compétition entre enseignes marchandes devient chose courante. La concurrence étant de plus en plus rude, les gestionnaires (responsable de magasin, chef de vente,...) sont obligés de maintenir l'attractivité et la rentabilité de leurs espaces de vente au risque de disparaître. La recherche de nouvelles stratégies et de nouvelles méthodes de merchandising est permanente. Cette recherche passe, en partie, par une meilleure connaissance du comportement des acheteurs. Dans les sous-sections suivantes, nous mettons en relief quelques méthodes et travaux en lien avec l'étude et la simulation de comportement de consommateurs dans le contexte du commerce.

1.2.1 Outils de simulation basés sur des comportements abstraits

Créer des populations de consommateurs virtuels pour étudier les effets de certaines stratégies de marketing est une solution qui est de plus en plus pratiquée. Le rôle de la simulation intervient, justement, pour reproduire une large panoplie d'individus virtuels aux actions, motivations et attitudes différentes. D'où la nécessité d'étudier les comportements des consommateurs. Ces derniers interagissent entre eux et agissent selon différents processus tels que l'imitation, la méfiance, le désir, la tentation... De même, le choix d'un consommateur est très souvent le fruit des influences des marques, de la publicité et du bouche à oreille... Les outils de décision marketing traditionnels donnent peu de descriptions opérationnelles des processus cognitifs. De plus, ces outils ne prennent pas en considération la dynamique sociale résultante de l'interaction dans le marché réel. Enfin, ils sont mal adaptés à une simulation reproduisant plusieurs milliers de consommateurs. Afin de pallier ces difficultés, des travaux se sont basés uniquement sur la représentation cognitive abstraite des consommateurs. Ils se concentrent sur l'entre-jeu entre les comportements individuels, les profils socio-économiques et les règles d'interaction.

Les outils développés à cet effet permettent la simulation de :

- l'attitude (comportementale) du consommateur ;
- l'impact des actes de consommation résultant de ces attitudes ;
- la rétroactivité des effets de ces actes sur les consommateurs eux-mêmes ;
- la réaction de la marque aux évolutions de marché et leurs effets rétroactifs sur les attitudes comportementales des individus.

Une des spécificités de ce type d'outil est qu'il prend en compte non seulement les caractéristiques cognitives individuelles, mais également les interactions entre consommateurs. Ces outils sont généralement adaptés lorsque l'on cherche à mieux comprendre l'impact d'une marque, d'un produit sur une masse de consommateurs et notamment via l'influence des prescripteurs au sein d'une communauté ou d'un réseau social.

A une échelle plus restreinte, en l'occurrence l'étude de comportements de consommateurs dans un espace de vente, l'analyse des stratégies commerciales ne dépend plus entièrement de la renommée de l'entreprise, des produits, mais de l'agencement et de l'ergonomie de l'espace [Bonnin, 2003].

Une première investigation [Terano *et al.*, 2009] basée sur des comportements abstraits de clients a été réalisée afin d'étudier le déplacement des clients dans un supermarché. Ces derniers sont conçus sous la forme d'un modèle conceptuel regroupant une liste d'achat, une probabilité d'achat visuel (par

pulsion), un ratio de reconnaissance de produits promotionnels s'ils se trouvent à portée de vision... L'agent est aussi doté d'un comportement tel que son déplacement soit uniquement influencé par la liste de produits, planifiée à l'avance, et l'agencement intérieur du magasin. Les résultats de la simulation, ainsi paramétrée et exécutée après avoir conçu l'agencement initial du magasin, coïncident avec les données réelles. Cet outil de simulation permet ainsi d'effectuer des expériences virtuelles en réalisant des changements sur les différents paramètres des opérations commerciales et la disposition des rayons [Terano *et al.*, 2009].

Cela étant, il est néanmoins nécessaire de préciser que plusieurs paramètres d'initialisation de la simulation sont intuitifs tels que le champ de vision des personnes (qui varie considérablement d'une personne à une autre), la liste d'achat initiale dont disposent les agents (certains clients n'ont pas forcément une liste préétablie ou même l'intention d'acheter)...

De plus, bien que la liste des produits achetés par chaque client permette de donner une certaine idée sur le parcours des consommateurs, ces derniers peuvent se déplacer à travers plusieurs rayons sans pour autant saisir des produits. Cette information est utile et devrait être prise en compte afin d'obtenir une meilleure connaissance sur ce qui peut intéresser éventuellement les consommateurs et cela en complément des achats effectivement réalisés.

Dans [Trakulpipat et Sinthupinyo, 2008], les auteurs présentent également une simulation à base d'agents d'un magasin. Leur contribution porte sur l'amélioration du couple (magasin, client) simulé. Comme précédemment, la modélisation des comportements des clients reste abstraite et ne s'appuie pas sur l'observation de données réelles : chacun des agents possède une liste de courses et procède selon un processus de décision simple pour choisir le produit désiré et le chemin à prendre, jusqu'à épuisement de la liste. L'originalité de l'article réside dans le calcul d'une valeur de satisfaction pour chaque agent. Ceci permet d'inférer une quantification et une approximation de la "désirabilité" de l'agencement du magasin. Cette approximation est faite pour chaque agencement qui sera ajusté par un algorithme génétique afin de s'adapter aux comportements des clients [Trakulpipat et Sinthupinyo, 2008].

Dans cette étude, l'agent est caractérisé par l'un des états suivants : chercher une cible, atteindre la cible et réussir l'achat. Pour chaque état, il existe une description de la motivation et du comportement de l'agent ainsi que les actions qui lui sont associées. L'objectif de l'agent est de maximiser la valeur de satisfaction qui dépend de l'achat des produits. Ces processus d'achat diffèrent selon l'agent : respecter une liste d'achats, parcourir les rayons avant de prendre une décision d'achat ou acheter par impulsion. L'idée est donc de faire une approximation de l'attractivité de chaque linéaire par rapport à la valeur de satisfaction des agents. Le fonctionnement de l'algorithme génétique se base sur des cycles itératifs d'évaluation (la fonction de satisfaction), de sélection (choix aléatoire de couples d'individus dits chromosomes sur toutes les données) et de mutation ou croisement (où chaque couple permet d'obtenir 2 nouveaux chromosomes fils, ayant subi une mutation dans une de leurs caractéristiques, ou ayant échangé une portion du chromosome). L'application de cette méthode pour l'amélioration de la conception du magasin nécessite un codage de chromosomes. Chaque chromosome représente une catégorie de produits dans un rayon et les coordonnées (X,Y) de sa position dans le magasin, voir figure 1.1.

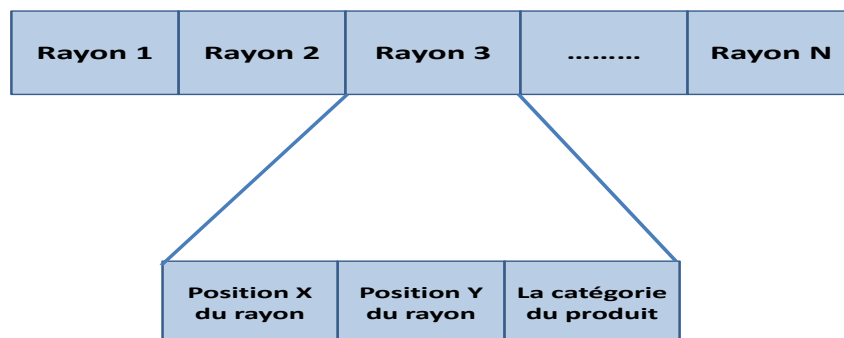


FIGURE 1.1 – Méthode de codage du chromosome

Vu la nature itérative de l'algorithme, l'effet de la mutation et du croisement, les chromosomes survivants dans la nouvelle génération représentent ceux qui décrivent la conception de l'espace ayant eu la valeur de satisfaction la plus élevée.

Les résultats expérimentaux de la simulation ont montré que cette méthode permet la découverte du meilleur aménagement possible. Cependant, ce résultat est bâti sur des hypothèses fortes et une modélisation des clients dont la pertinence et le réalisme restent à vérifier. Il est aussi important de signaler que la mesure quantitative de la satisfaction des clients reste subjective et ne peut pas correspondre à la réalité.

D'une façon générale, ces études basées sur des comportements abstraits requièrent une première phase qui nécessite de faire appel aux connaissances d'un expert afin de récolter quelques informations utiles à la conception et la modélisation des clients virtuels. De fait, l'intervention de l'expert peut être vue comme un inconvénient. En effet, cela réduit la généralité ou la portabilité du produit qui ne va être effectif que pour le magasin en question. La totalité du processus de développement de l'outil doit être reprise si un manager d'un autre magasin souhaite exploiter l'outil de simulation.

Dans ces travaux, les comportements des individus au sein des simulations sont construits sous la base d'hypothèses fortes et difficilement vérifiables. Ainsi, la génération de certaines données est très aléatoire (par exemple les listes d'achats que possède chaque individu virtuel, la motivation pour un produit,...). D'autres travaux, qui seront détaillés dans la section suivante, exploitent directement des comportements de clients réels.

1.2.2 Outils de simulation basés sur des comportements de clients réels

Dans le contexte de simulation de magasin de vente en libre-service, de nombreuses études s'appuient sur des modèles basés sur une recherche multidisciplinaire portant sur l'étude économique, psychologique et informatique d'une population de clients [Siebers *et al.*, 2008b] [Siebers *et al.*, 2008a]. Les données utilisées dans ce type de modélisation proviennent d'enquêtes consommateurs (observations informelles, interviews et questionnaires) et de documents liés à la gestion et l'exploitation du point de vente. A l'instar des outils présentés précédemment, les modèles obtenus permettent de :

- fournir un support de décision pour la gestion du magasin ;
- évaluer l'agencement et la conception du magasin.

Seule diffère leur mise en place, qui nécessite une phase d'étude préalable concernant la représentation des comportements des consommateurs. Une étape de collecte de données est donc nécessaire et pour cela des questionnaires exhaustifs et précis doivent être réalisés. Parmi les autres difficultés, il est à noter que ce type d'enquête repose sur le déclaratif des personnes interviewées ; ce qui représente une source de biais pour la phase de modélisation.

D'autres travaux portent également sur les enquêtes de consommateurs, mais d'un point de vue différent, en s'appuyant sur l'analyse des parcours et des flux au sein du magasin. Cette méthode a été établie dans le cadre d'une étude visant à évaluer des stratégies concurrentielles permettant d'améliorer la qualité de service des supermarchés [Teknomo et Gerilla, 2009]. Elle porte sur la collecte de données relatives aux déplacements des clients, sur leur analyse et leur modélisation. La collecte de ces données consiste en un simple inventaire des produits ainsi que des enquêtes par questionnaire permettant de déduire la trajectoire de déambulation des clients. La technique utilisée se base ensuite sur l'analyse de trajectoires statiques obtenues à partir des données collectées afin d'en faire ressortir une agrégation des schémas de flux. Cette information permet d'apprendre quelles allées reçoivent plus de clients que d'autres et quels produits sont préférés à d'autres et, par conséquent, elle permet d'avoir une connaissance sur l'attractivité des rayons et des produits.

Certaines sociétés technologiques proposent des offres commerciales pour des outils de supports de gestion de magasin se basant sur l'étude des comportements des clients. A titre d'exemple, un concept de simulation de magasins a été développé par la société ACTISKU¹, spécialiste dans les applications de conception de magasins virtuels 3D. Cet outil représente une surface de vente en 3D et permet de reproduire une réplique exacte d'un magasin de vente réel. Il va permettre aux cyber-consommateurs de circuler de façon virtuelle dans le magasin et de pouvoir manipuler et acheter les produits désirés. ACTISKU fournit aussi des résultats qui vont permettre d'étudier l'impact de certaines stratégies de vente analysant le comportement des consommateurs. Notamment, les points chauds et les points froids du magasin, les produits manipulés, le sens de circulation... Dans ce cas, le recueil de données repose sur les traces des consommateurs virtuels dont les comportements ne peuvent pas correspondre exactement avec ceux qui se déplacent réellement dans un vrai magasin du fait d'une immersion limitée au visuel (pas de sollicitation des autres sens : tactile, olfactif...).

Un autre outil nommé TC-Store² a vu le jour, créé par une équipe de professionnels spécialisée dans la conception et le développement de solutions de retail intelligence³. L'outil est conçu pour accroître la rentabilité de l'entreprise à partir d'une meilleure connaissance du comportement des clients. Pour ce faire, le système proposé repose sur des équipements tels que compteur de personnes et détecteur de mouvement, ainsi qu'un logiciel de gestion spécifique. Cette combinaison de matériels et de logiciel est, par conséquent, capable de mesurer certains paramètres tels que le nombre d'entrées des clients

1. <http://www.actisku.com/>

2. <http://www.t-cuento.com/fre/tc-store.php?x=Qu%E9+es+TC-Store>

3. Le Retail Intelligence représente l'ensemble des outils et des applications qui ciblent la création et la gestion de connaissances obtenues grâce au recueil et à l'analyse des chiffres dans une organisation ou dans une entreprise de vente au détail.

dans l'établissement, la capacité d'occupation du magasin dans chaque période, la durée de l'acte de vente, ainsi que les flux de piétons qui passent chaque jour devant le magasin. Cependant, les données recueillies, relatives aux comportements ; à part le fait qu'elles puissent répondre à certaines questions liées au potentiel des produits ou l'attractivité des vitrines ; ne permettent pas de suivre individuellement les clients et distinguer leurs différents profils. Cette connaissance peut s'avérer utile pour prendre en compte la diversité des goûts et adapter le magasin autant que possible aux différentes attentes des clients.

En dépit du contexte virtuel de ACTISKU et le besoin d'installation de matériels spécifiques et coûteux de TC-Store, ces types de logiciels commerciaux présentent un certain nombre de caractéristiques intéressantes. En effet, un des points positifs de ces outils est que le recueil et l'analyse des données du comportement des consommateurs se fait en quelque sorte de façon automatique. Il n'est pas nécessaire, dans ce cas, d'avoir recours à un expert du domaine et faire appel à ces connaissances et son expertise pour aboutir à l'analyse des comportements.

La méthodologie, que nous proposons dans cette thèse, s'oriente vers une simulation comportementale réelle basée sur les systèmes multi-agents. Les travaux de [Teknomo et Gerilla, 2009], [Siebers *et al.*, 2008b], [Terano *et al.*, 2009] et [Trakulpipat et Sinthupinyo, 2008] se basent, d'ailleurs, sur cette technologie. La section suivante est dédiée, de ce fait, aux systèmes multi-agents.

1.3 Les systèmes multi-agents (SMA) et simulation

L'emploi des SMA varie selon les besoins des applications. Ils peuvent être utilisés dans un cadre de coopération et de négociation, pour effectuer des stratégies de coopération entre entités et pour résoudre des problèmes distribués [Alshabi *et al.*, 2007]. Nous trouvons également la notion de coopération entre des robots formant un SMA tels que les robots footballeurs par exemple [Kim *et al.*, 1997].

D'un autre côté, les SMA peuvent s'appliquer dans un cadre compétitif [Tuyls *et al.*, 2006]. Ils peuvent aussi être efficaces dans un contexte de contrôle [Huberman et Clearwater, 1995], de diagnostic de défaillances [Jennings et Bussmann, 2003], ou celui de l'évolution de réseaux sociaux [Amblard et Ferrand, 1998],...

Une liste exhaustive des apports et des innovations relatifs à cette technologie serait ambitieuse. Toutefois, nous allons porter notre attention sur un type particulier de SMA qui permet la reproduction d'un phénomène donné du monde réel au moyen d'une simulation. Nous nous intéressons à cela afin de comprendre ou expliquer un comportement collectif et l'utiliser pour inférer une connaissance utile.

Dans ce qui suit, nous dressons tout d'abord une définition du concept de système multi-agents, en partant du composant élémentaire : l'agent. Ensuite, nous abordons les simulations multi-agents à travers un bref historique des travaux effectués dans le domaine.

1.3.1 L'agent et ses caractéristiques

Il existe plusieurs définitions du concept "agent" en raison de la diversité des entités qu'il peut désigner. La définition la plus universelle précise qu'un agent représente une entité qui agit. De façon plus précise et selon Jacques Ferber [Ferber, 2007], un agent est une entité (software ou hardware) située dans un environnement (virtuel ou réel) qui :

- est capable d'agir dans un environnement ;
- est conduit par un ensemble de tendances (buts) ;
- possède ses propres ressources ;
- peut communiquer avec les autres agents ;
- possède un comportement autonome dépendant de ses perceptions, ses représentations et ses interactions avec l'environnement et les autres agents ;
- a une représentation partielle de l'environnement où il évolue ;
- peut éventuellement se reproduire.

En d'autres termes, un agent est capable de raisonner, représenter, communiquer et décider. Il peut manifester différentes propriétés [Axtell, 1999] telles que :

- l'autonomie : ne dépend d'aucune autre entité ;
- la sociabilité : prend en compte les autres agents et interagit avec eux ;
- la réactivité : construit sur un cycle de perception-réaction ;
- la pro-activité : possède des motivations et réagit à son environnement selon les buts à atteindre ;
- la localité : possède une limitation à sa perception et à ses actions.

A quelques divergences près, les définitions des agents se rejoignent toutes. Les quelques divergences concernent le type d'applications pour lesquelles est conçu l'agent. D'où l'existence de plusieurs types d'agent.

1.3.2 Les types d'agents

Parmi les différents types d'agent que nous pouvons rencontrer, il en existe trois considérés comme principaux :

- Réactif : lorsque l'agent n'a qu'une faible capacité de communication (par exemple, une communication indirecte via des traces laissées dans l'environnement) et pas de représentation explicite de l'environnement auquel il appartient. Il possède une liste d'objectifs et dispose d'un comportement de type stimulus-réponse en réaction à ce qu'il perçoit ;
- Cognitif : il possède un objectif à atteindre, un plan à réaliser et des capacités de mémoire et de raisonnement importantes. Il est donc capable de prendre des décisions évoluées et peut procéder par communication intentionnelle (échange de messages) ;
- Social : qui tient compte des autres agents et participe à l'accomplissement de leurs tâches quand la situation le réclame et lui est favorable.

Ces agents sont destinés à interagir avec d'autres agents dans un environnement formant, donc, un SMA. L'environnement permet aux agents d'évoluer, de communiquer, de percevoir et d'agir. Ces propriétés font du SMA, un système qui se distingue des autres systèmes classiques, notamment les systèmes distribués et les systèmes orientés objets.

Dans notre cas d'étude, l'objectif recherché consiste à reproduire les comportements des consommateurs dans un magasin. Il convient donc d'utiliser des agents plutôt réactifs avec des capacités de raisonnement plutôt simples. L'agent est ainsi doté d'un ensemble de buts à atteindre (les produits recherchés). Il doit éviter les obstacles, rayons et mobilier du magasin par exemple, y compris les autres agents (consommateurs, vendeurs,...). Ces interactions provoquées par les comportements des agents entraînent un comportement global émergent, phénomène couramment rencontré dans les applications de simulation. Ainsi, nous nous intéressons, dans ce qui suit, aux SMA réactifs, aux simulations et aux phénomènes émergents.

1.3.3 Les SMA et leurs caractéristiques

De nombreuses communautés scientifiques, ainsi qu'industrielles, font usage du paradigme agent pour répondre à des problématiques issues de vastes champs d'application. Ceci est dû en partie à la transversalité des SMA. En effet, les systèmes multi-agents permettent de représenter des systèmes dynamiques constitués de composants ou d'entités dotés de comportements différents. Les SMA sont ainsi largement appliqués dans les systèmes répartis, les jeux, la robotique autonome et communicante, les réseaux de communication, les processus cognitifs et d'apprentissage, l'imagerie, l'analyse de scènes, l'analyse de comportements, ...

Un SMA est caractérisé principalement par :

- L'intelligence collective des agents : résultante des interactions entre des agents autonomes, dotés de capacités cognitives et d'un comportement réflexif leur permettant d'agir, d'observer leurs actions et d'en changer le cours, ce qui pourrait être beaucoup plus complexe voir inconcevable s'ils étaient isolés ;
- L'interactivité des agents : situés dans un environnement partagé et ouvert, les agents peuvent interagir en communiquant entre eux ou en agissant sur l'environnement ;
- La complexité du système : qui peut contenir de nombreux agents hétérogènes ;
- L'ouverture : qui désigne l'échange d'information avec l'extérieur, ainsi que l'évolution fonctionnelle du système relative à la mobilité des agents, leur entrée et leur sortie ;
- La distribution : qui concerne la modularité des connaissances et des traitements ;
- L'adaptation : les agents tiennent compte de la dynamique de leur environnement pour ajuster leur comportement en cours de fonctionnement ;
- L'intelligibilité : dénote le caractère accessible et interprétable du SMA offrant un ensemble d'abstractions modélisées qui correspondent, de manière assez naturelle, au système cible.

1.3.4 Les SMA réactifs

Parmi les modèles classiques des SMA, l'approche réactive a fait l'objet d'un bon nombre d'applications. Cela est dû à ses propriétés intrinsèques : simplicité, flexibilité et fiabilité, à sa capacité d'adaptation. Aussi, elle a su répondre efficacement aux exigences des problèmes complexes, notamment, dans des problèmes d'évitement d'obstacles pour des systèmes d'aide à la conduite [Gechter *et al.*, 2010], dans la résolution de problèmes [Simonin et Gechter, 2005], dans les simulations de modèles de sociétés telles que les colonies de fourmis [Drogoul et Ferber, 1994],...

Un SMA réactif évolue dans le temps et peut engendrer des phénomènes émergents résultant des comportements individuels des agents et de la dynamique propre à l'environnement. Dans un SMA

réactif, l'agent ne possède pas de représentation explicite de l'environnement et de l'organisation dont il fait partie. Il est considéré comme un agent "simple" dépourvu de raisonnement abstrait et complexe. Les activités et l'interaction des agents induisent l'organisation du système.

Malgré leurs nombreuses qualités, les SMA réactifs soulèvent néanmoins, tout comme n'importe quel type de SMA, beaucoup de difficultés et de questions intrinsèques à leur conception [Iglesias *et al.*, 1997] à savoir :

- Comment décrire, décomposer et formuler un problème à résoudre en agents ?
- Comment spécifier les caractéristiques d'un agent ?
- Comment décrire la connaissance requise par les agents pour atteindre leurs objectifs ?
- Comment concevoir les outils et les méthodologies de développement pour les SMA ?

Ces questions seront abordées et traitées ultérieurement dans notre méthodologie, tout au long des chapitres.

1.3.5 Les simulations multi-agents

Les simulations multi-agents s'avèrent être un "micro monde artificiel" dont les caractéristiques peuvent être contrôlées et permettent de reproduire une série d'expériences comme dans un laboratoire. Les simulations à base d'agents représentent aujourd'hui une alternative crédible aux approches classiques permettant de modéliser des systèmes complexes, notamment lorsque ces systèmes sont difficilement représentés sous forme d'équations.

Ces simulations consistent à utiliser un ensemble d'agents en interaction de manière à reproduire la dynamique et l'évolution des phénomènes que l'on cherche à simuler. La simulation à partir d'une modélisation "agent" permet donc de s'affranchir des limites des approches analytiques et de créer un paradigme puissant permettant d'aborder l'étude et l'analyse de systèmes réels complexes.

1.3.6 L'émergence dans les simulations multi-agents

L'émergence au sein d'un SMA caractérise les phénomènes qui apparaissent de manière globale au sein du système à partir des interactions locales entre agents. En ce sens, l'émergence peut être vue comme un passage du niveau micro au niveau macro. Le processus de modélisation d'une simulation multi-agents, partant d'une observation et entraînant un résultat émergent, est illustré dans la figure 1.2.

Le concept de l'émergence a fait l'objet de nombreuses études dont celle décrite dans [Müller, 2003] ainsi que dans [Georgé *et al.*, 2003] représentant une théorie sur les systèmes multi-agents adaptatifs à fonctionnalité émergente. Le comportement collectif émergent peut induire à une connaissance pertinente. Néanmoins, la séquence d'événements qui produit le phénomène émergent est cachée. Lors de la construction d'un SMA, les entités qui le composent ainsi que leurs relations peuvent être contrôlées et maîtrisées. Cependant, le comportement émergent de ces composants est implicite et imprévisible. Il n'existe aucune certitude à atteindre un comportement global désiré. Il peut, toutefois, conduire à l'apparition de structures, de motifs ou de propriétés nouvelles et cohérentes [Jeffrey, 1999]. Cette notion d'émergence est également exploitée dans différents travaux portant sur les méthodes de conception. Dans ce qui suit, nous examinons de plus près certaines études qui traitent des méthodologies portant sur notre problématique principale qui est la conception de simulation multi-agents dont certaines s'appuient sur le principe d'émergence.

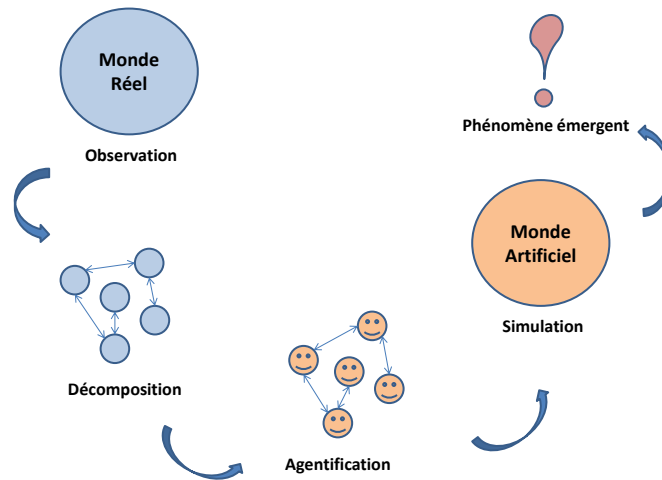


FIGURE 1.2 – Processus de modélisation de simulation multi-agents

1.4 Méthodes de conception de simulation multi-agents

Dans cette section, et dorénavant tout au long de ce manuscrit, nous nous focalisons sur la problématique de modélisation et de conception de simulations de comportements basées sur les systèmes multi-agents. Nous proposons d'exposer différentes architectures existantes de simulation de comportements observés. Nous soulignons pour chacune leurs points forts et leurs points faibles dans le processus de modélisation.

La simulation multi-agents part généralement d'une approche descendante consistant à identifier d'une part les entités composant le phénomène réel à reproduire et d'autre part, les relations entre ces entités. Chaque entité est ensuite modélisée sous la forme d'un agent, généralement en faisant quelques hypothèses simplificatrices. Comme toute approche de modélisation, les modélisations "agents" restent donc une approximation ne permettant pas de prendre en considération tous les détails du ou des phénomènes réels à simuler. Dans certains cas, cette approximation peut conduire à des écarts importants entre le réel et le simulé induisant ainsi des interprétations, des analyses et des résultats erronés. Il existe donc une vraie problématique autour de la caractérisation des entités essentielles à modéliser et de la détermination du niveau de détail de leur modélisation.

Comme mentionnées auparavant, certaines approches utilisent le concept de l'émergence dans la conception d'un SMA. L'auteur dans [Klein, 2009] a regroupé ces travaux selon deux aspects différents : les approches par construction et les approches par contrôle (le terme contrôle qui est utilisé par cet auteur décrit un moyen d'agir sur un SMA pendant son exécution en fonction de son évolution courante). Il s'agit en fait de démarches au niveau micro et de démarches au niveau macro, respectivement, pour spécifier le comportement collectif d'une simulation SMA. Ces moyens de conceptions sont présentés ci-après, suivis de deux autres méthodes récentes et originales de conception de simulation multi-agents. La première fait participer des agents humains capables d'agir

sur le comportement des agents en cours d'exécution et, par conséquent, sur le système global. La seconde repose intégralement sur les interactions entre les agents impliqués dans la simulation.

1.4.1 Démarche au niveau micro

La démarche au niveau micro (ou approche par construction) permet de s'assurer que le SMA présente un comportement spécifié lors de sa construction. On en distingue deux types décrits dans les deux paragraphes ci-après :

1.4.1.1 Approche par méthodes de conception

Cette approche est utilisée lorsqu'on définit entièrement le comportement des agents, sans cadre formel qui délimite leurs capacités. Un intervenant expérimenté, appréhendant le comportement global du SMA, détermine et exerce une correction des comportements des agents. Les auteurs dans [Wolf et Holvoet, 2005] et dans [Di *et al.*, 2006] préconisent cette méthode en s'appuyant sur l'idée que la seule possibilité d'obtenir un comportement cohérent au niveau macro est de laisser le comportement se produire et s'organiser de façon autonome, ce qu'ils appellent l'auto-organisation.

Les méthodes de conception, bien qu'elles puissent être efficaces dans certains cadres d'application, demeurent très dépendantes de l'intervention d'un humain qui va non seulement déterminer mais aussi corriger les comportements des agents. Ceci peut être problématique dans certains cas. Notamment dans le cas où le concepteur fait un mauvais choix initial. Il est alors nécessaire d'effectuer un nombre important de tests et de calculs pour n'omettre aucune amélioration possible.

1.4.1.2 Approche par ajustement des paramètres

Cette approche représente une optimisation empirique sur l'espace des paramètres du SMA à calibrer, en explorant et en retenant la meilleure combinaison de valeurs. Quelques techniques proposent d'utiliser la connaissance du système réel pour scinder le modèle général en sous-modèles et les traiter individuellement par un processus de calibration [Fehler *et al.*, 2004]. Ceci peut s'avérer pertinent dans le cas où le nombre de paramètres du système multi-agents à ajuster est élevé ; les dépendances des paramètres sont complexes à cause des interactions entre les agents ; ou le coût du calcul de la simulation est très élevé. D'autres techniques sont également utilisées pour l'exploration de l'espace des paramètres tels que les algorithmes génétiques, par exemple, qui cherchent à parcourir les sous-espaces susceptibles de fournir les meilleurs résultats selon une valeur globale directement mesurable appelée "fitness" (ou valeur de satisfaction comme mentionnée auparavant dans le paragraphe 1.2.1).

Dans le cas d'une simulation reproduisant un espace de vente, s'assurer de concevoir d'emblée des comportements d'agents (consommateurs, vendeurs, caissiers, produits,...) convenables et satisfaisants, que ce soit en appliquant les méthodes de conception ou les méthodes par ajustement de paramètres, peut paraître excessivement ambitieux. En effet, faire explorer l'espace des paramètres de la simulation et des agents par un expert serait subjectif et inapproprié pour refléter la réalité.

1.4.2 Démarche au niveau macro

Le principe de cette approche repose sur l'influence dynamique de l'utilisateur sur un SMA existant, afin de diriger un comportement global vers un comportement désiré. Le but est de pouvoir agir pendant l'exécution du système afin de modifier le comportement collectif. Le recours à ces méthodes est motivé généralement par l'échec des méthodes par construction (niveau micro).

La mise en place de cette approche est similaire au contrôle d'un système dynamique en automatique qui nécessite que le système soit muni :

- D'une cible : l'ensemble des comportements désirés ;
- De moyens d'observation : moyens pouvant fournir des indications sur l'état du SMA ;
- D'un ensemble de moyens d'action sur le SMA.

Pour assurer le succès de la démarche, il faut que le phénomène émergent ait un effet correcteur, par rétroaction, sur les entités qui composent le système. Or, la difficulté intrinsèque aux SMA est qu'on est incapable de prédire l'effet qu'une modification du système ou des agents peut provoquer sur le comportement global. Déterminer quelle modification est nécessaire pour atteindre un comportement global souhaité est souvent compliqué.

De ce fait, des approches de vérification d'un SMA sont nées. Parmi ces approches nous citons :

- Utiliser l'architecture observateur/contrôleur [Richter *et al.*, 2006] où les agents sont dotés d'un comportement autonome et sont capables de s'adapter à leur environnement. Le comportement présumé émergent est ensuite mesuré au moyen d'un observateur, qui va fournir une description abstraite de l'état courant du SMA au contrôleur. Ce dernier analyse l'observation obtenue et dans le cas où cette description est non satisfaisante (éloignée du comportement cible), il va réagir en déterminant les actions, appelées actions de contrôle, qui vont guider l'organisation des agents en influant sur certains paramètres locaux.
- Représenter chaque agent par un MDP (Processus de Décision Markovien) [Scherrer, 2004] devant optimiser sa propre politique, lui-même étant dépendant de la politique des autres agents, selon un objectif ciblé. Le critère d'optimalité (la fonction de récompense dans les MDP) est global, et correspond au comportement du SMA que l'on cherche à atteindre. Il peut néanmoins être décomposé en une somme de fonctions de récompenses locales [Chadès *et al.*, 2002]. Cette approche est généralement utilisée dans un problème de collaboration.
- Appliquer le contrôle au niveau global en deux étapes. La première représente une phase d'apprentissage qui va construire un modèle de la dynamique globale du SMA, sous forme de graphe d'états, de façon expérimentale. La deuxième repose sur l'exploitation du modèle créé et le choix des actions de contrôle de façon rationnelle [Klein *et al.*, 2009].

Ces méthodes, permettant d'utiliser l'information du niveau macro au niveau micro, semblent être prometteuses. Ces procédés sont réalisés de façon automatique, ce qui rend ce traitement plus objectif et limite le recours au jugement humain. Cependant, déterminer les informations globales appropriées qui vont être utilisées pour agir sur le système est sujet à discussion. La difficulté réside dans le choix des paramètres au niveau micro qui vont être affectés et altérés par les actions du "contrôleur". Malgré le faible nombre d'études concernant ces méthodes, il pourrait être envisageable de les étudier dans notre cadre applicatif : la simulation d'un espace de vente, et de l'intégrer dans notre méthodologie que nous présentons ultérieurement.

Néanmoins, ces approches ne peuvent être mises en pratique que sur un SMA existant afin d'y apporter des améliorations pour le conduire vers un comportement désiré. Elles représentent donc un moyen d'appui lors de la conception de simulation agent et non pas des méthodes de conception proprement dites. Ce qui nous amène à prospecter d'autres pistes de conception de SMA.

1.4.3 Conception de simulation participative

Parmi les méthodes facilitant la conception de simulation se trouvent les simulations multi-agents participatives [Guyot et Honiden, 2006]. Celles-ci sont bien sûr basées sur le paradigme agents, cependant, un nouveau concept fait son apparition dans le système : la participation des humains. Ceux-ci sont autorisés à agir sur le système de la même manière que le ferait un agent. Cette technique suit un diagramme itératif (voir figure 1.3) qui va transformer successivement le modèle, la définition de primitives et l'interface établis. Sa conception est spécifique au système ciblé qu'on cherche à simuler et commence par une phase d'observation et de collecte de données qui va aider à établir la structure du modèle (par exemple la description des interactions qui se déroulent entre des acheteurs et des vendeurs d'un produit donné : protocole d'interaction). Ensuite vient la conception du modèle qui va établir une description formelle des éléments du modèle en ayant recours à l'aide des spécialistes du domaine le cas échéant. Ces descriptions formelles des interactions entre les agents se présentent sous forme de primitives (actions primitives).

A partir de là, le modèle de la simulation prend forme en tenant compte des contraintes et des buts de l'expérimentation recherchée. Les participants contrôlant chaque agent sont capables de redéfinir leurs rôles et les règles sociales en utilisant les primitives qui leur permettent de suivre librement n'importe quelle stratégie. Ces actions sont ensuite enregistrées et traitées durant la session de débriefing afin que les participants soient confrontés à une vision globale de leurs comportements.

Ce mécanisme permet d'accomplir des expériences, en impliquant des intervenants humains. Il permet, de ce fait, de concevoir des systèmes multi-agents capables de s'améliorer et de s'adapter par rapport aux participants ; pouvant ainsi être profitable dans un contexte d'apprentissage et d'éducation. De plus, ce mécanisme pourrait être utilisé pour concevoir des protocoles d'interaction efficaces en tant que support de négociation. Ces simulations constituent une alternative crédible dans certaines applications, non seulement pour l'étude de comportements collectifs, mais également pour ses qualités pédagogiques et d'aide à la décision. Néanmoins, il dépend fortement de la qualité des interventions et n'est applicable que sur des applications nécessitant de fortes interactions.

1.4.4 Conception de simulation basée sur les interactions

La méthodologie IODA (Interaction-Oriented Design of Agent simulations) [Kubera *et al.*, 2011] [Mathieu *et al.*, 2001], facilite la conception d'une simulation basée sur des agents situés⁴. Cet outil se base sur les interactions des entités qui composent la simulation, dans le but d'obtenir un processus de conception de simulations entièrement orientées interactions. De ce fait, un agent est considéré comme une entité qui effectue ou subit un ensemble d'interactions sans pour autant être conçu spécifiquement pour réaliser un comportement donné.

4. Agents situés dans un environnement sur lequel ils peuvent effectuer des actions qui sont susceptibles de le changer.

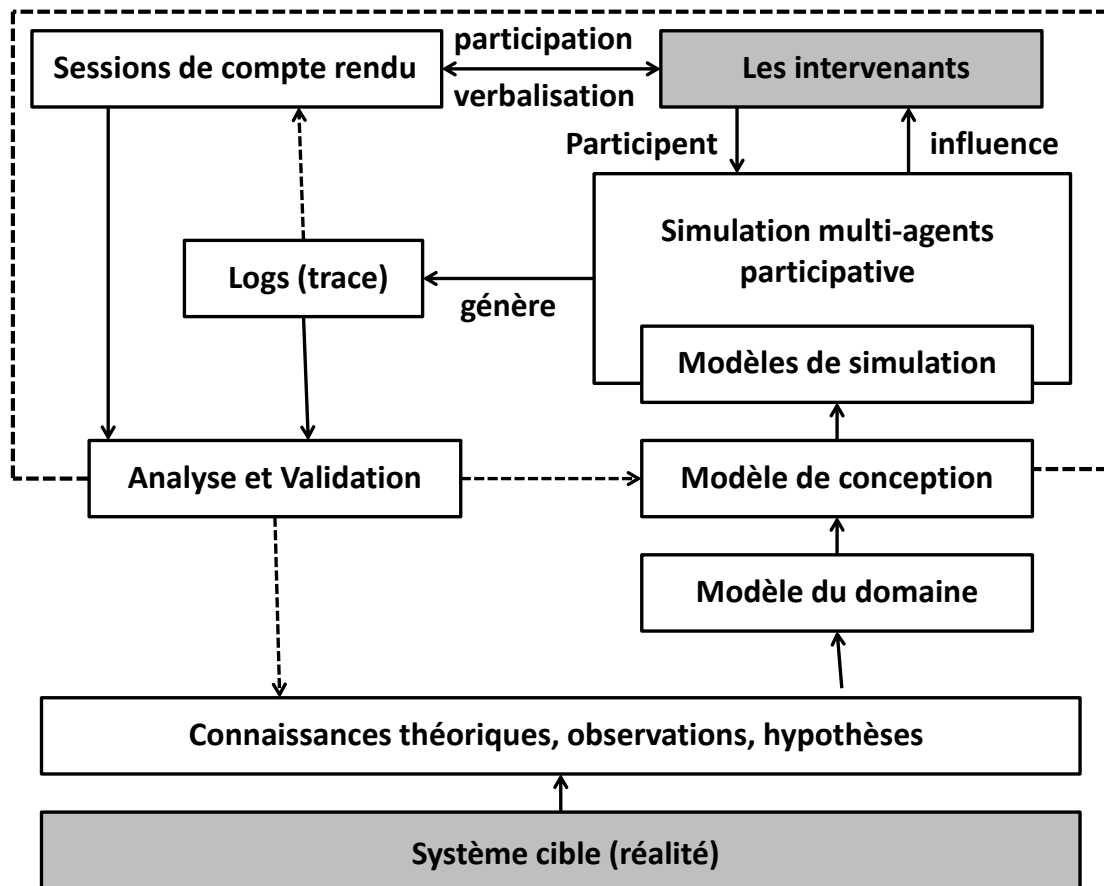


FIGURE 1.3 – Processus de simulation agent dans un contexte participatif [Guyot et Honiden, 2006]

L'accent est mis sur la modélisation des interactions au travers la définition d'une matrice d'interaction entre entités. Ceci rend la conception du système plus accessible, étant donné que la phase de conception des agents s'en trouve simplifiée. Parmi les informations que doit traiter le modèle formel de IODA, nous citons :

- les interactions qui représentent l'action choisie à partir de préconditions (expression booléenne décrivant les conditions logiques ou physiques nécessaires pour lancer l'interaction) et d'un élément de déclenchement (une expression booléenne décrivant l'objectif implicite ou explicite de l'interaction).
- la typologie d'interaction qui représente la relation entre un agent source et un agent cible selon une cardinalité d'interactions ;
- la distance limite qui sépare deux agents, car un agent ne peut interagir qu'avec ceux qui sont suffisamment proches ;
- les caractéristiques de l'environnement qui doit contenir les agents et qui permet le calcul des distances entre les agents et de leurs cercles de perception ;
- le fonctionnement d'un agent qui tient compte d'un ensemble d'interactions réalisables et d'un cercle de perception. Une sélection d'actions est ensuite réalisée puis l'état de l'agent est actualisé ;

- la matrice d'interactions qui vise à fournir un résumé de toutes les interactions se produisant dans la simulation.

Conformément à cette méthodologie, la conception d'une simulation est réalisée comme suit :

- Identifier des entités et assigner des interactions pour les agents sources et les agents cibles ;
- Définir et implémenter les primitives pour les conditions d'activation et les séquences d'actions pour les interactions pour chaque famille d'agent ;
- Définir pour chaque interaction I une priorité $p(I)$ et une distance limite $d(I)$;
- Définir comment évolue la matrice durant la simulation, dans le cas où l'agent peut modifier son comportement ou celui des autres, en agissant sur les cellules de la matrice.

La table 1.1 représente un exemple de matrice d'interactions d'une simple simulation d'un magasin, où la première colonne représente les agents sources et la première ligne représente les agents cibles. Les cellules non vides se trouvant dans l'intersection d'un agent source et un agent cible représentent l'interaction établie à une distance maximale d de la source.

Source \ Cible	∅	Consommateur	Vendeur	Produit	Caisse
Consommateur	(Se déplace) (Choisit produit) (Sortir)	(Evite, $d = 1$)	(Demande, $d = 0$)	(Prend, $d = 0$)	(Paye, $d = 0$)
Vendeur	(Se déplace)	(Aide, $d = 0$)	(Evite, $d = 1$)	(Remet, $d = 0$)	
Produit	(Se renouvelle)				
Caisse	(Encaisse)				

TABLE 1.1 – Matrice d'interaction d'une simulation d'un magasin de vente

Adopter cette méthodologie, centrée sur les interactions, permet de construire des comportements génériques pouvant être réutilisables dans différents contextes. Aussi, elle permet de séparer la connaissance sur les comportements du traitement. Ce qui donne plus de souplesse dans la conception des simulations. Cependant, toute simulation multi-agents ne repose pas seulement sur les interactions, ou du moins, l'interaction entre les agents ne représente pas toujours le facteur principal du système à simuler. Dans ces cas là, il est défavorable d'adopter cette méthodologie de conception.

L'ensemble des approches exposées dans cette section exige d'importants efforts en matière de modélisation et de programmation. Le concepteur de la simulation doit disposer d'une certaine expertise du système à simuler pour pouvoir le modéliser sous forme d'agents. Pour de nombreuses applications, l'observation du système à simuler et le recueil de données réelles sont possibles mais fastidieux à réaliser pour le concepteur. Ainsi, il est intéressant de s'appuyer sur des techniques, en l'occurrence la fouille de données ou Data Mining (DM), permettant d'automatiser la phase d'observation du système réel afin d'identifier ou valider les entités qui le composent et leurs comportements, et cela, en évitant une éventuelle subjectivité du concepteur ou de l'expert.

1.5 Combinaison du Data Mining et des simulations agents

Il existe, dans la littérature, certaines approches qui combinent le Data Mining avec les simulations à base d'agents, notamment dans les travaux de [Baquero *et al.*, 2009]. Cette association s'avère pertinente notamment grâce à la capacité des techniques d'analyse de données (ou Data Mining)

à acquérir automatiquement des descriptions structurales à partir d'exemples, pouvant améliorer le fonctionnement d'une simulation multi-agents et par conséquent ses résultats. Nous exposons, dans cette section, l'intérêt du Data Mining ainsi que les différentes façons de l'appliquer dans une simulation multi-agents.

Le Data Mining [Frawley *et al.*, 1992] représente l'étape de base du "Knowledge Discovery in Database" qui se définit comme le processus d'identification de modèles (patterns) valides, nouveaux, potentiellement utiles et compréhensibles à partir de données [Remondino et Correndo, 2006].

L'objectif du Data Mining est d'extraire, à partir d'un volume important de données, des connaissances ou des informations cachées, jusque-là inconnues, et susceptibles d'être pertinentes. Cette extraction se fait à travers la recherche d'associations, de tendances, de relations ou de modèles. Les techniques de Data Mining pour l'analyse des données ne traitent pas seulement des données structurées, mais aussi les données non structurées telles que les signaux, le texte, mais aussi les vidéos.

Dans [Remondino et Correndo, 2006], les auteurs ont divisé les tâches effectuées par le Data Mining dans les SMA suivant les deux catégories bien connues en analyse de données :

- les tâches descriptives qui permettent de découvrir des régularités pertinentes dans les données et de déceler des modèles cachés ;
- les tâches prédictives qui permettent de construire des modèles de prédiction, généralisables à des valeurs qui ne sont pas couvertes par les données initiales, comme dans le modèle d'association et le processus de classification.

Dans [Baqueiro *et al.*, 2009], deux approches intéressantes ont été exposées combinant le Data Mining et la modélisation orientée agent. La première porte sur l'utilisation des simulations à base d'agents pour améliorer un traitement de Data Mining. La simulation a pour but de fournir des données manquantes du système réel et ainsi améliorer le traitement de données, préalablement trop limitées. Le second adopte l'approche inverse où le Data Mining est utilisé pour améliorer une simulation à base d'agents et particulièrement pour aider à sa conception. La deuxième approche proposée consiste à utiliser le Data Mining pour améliorer la qualité de la simulation. Ce cas d'utilisation est principalement en lien avec la phase de vérification et de validation d'une simulation multi-agent et permet de pallier le manque de méthodologies de vérification et de validation dans les simulations.

Les interactions des entités dans les SMA peuvent aboutir à l'émergence de connaissances non-pertinentes ou implicites. Une phase supplémentaire peut être nécessaire dans l'espoir de révéler des modèles à partir des résultats de la simulation. Dans ce cadre, l'intégration du Data Mining au SMA permet de révéler des motifs pertinents et par conséquent de mieux comprendre les résultats de la simulation. Cette information permet d'agir sur la simulation en cherchant à l'améliorer. La vérification permet, en l'occurrence, de s'assurer que l'implémentation du modèle est effectivement une représentation correcte du modèle abstrait. Alors que la validation permet de vérifier si le modèle implémenté peut atteindre les objectifs proposés par les expériences de la simulation.

Parmi les techniques de Data Mining utilisées dans les processus de validation des simulations à base d'agents, nous pouvons citer la classification, le clustering, les règles d'association et l'identification de motifs (patterns) séquentiels à partir des données obtenues en simulation. L'utilisation

des techniques de Data Mining permet d'abstraire certains résultats expérimentaux et obtenir des éléments de description de haut niveau représentant le comportement observé dans les simulations [Remondino et Correndo, 2006]. Ceci permet une comparaison plus simple entre les résultats expérimentaux obtenus d'un même modèle ou entre des modèles indépendants représentant le même phénomène.

Le modèle de la figure 1.4 décrit le mécanisme de vérification et de validation en appliquant le Data Mining dans les simulations orientées agents. Cependant, cette intégration peut se réaliser de différentes manières que nous pouvons classer en deux catégories principales : endogène et exogène.

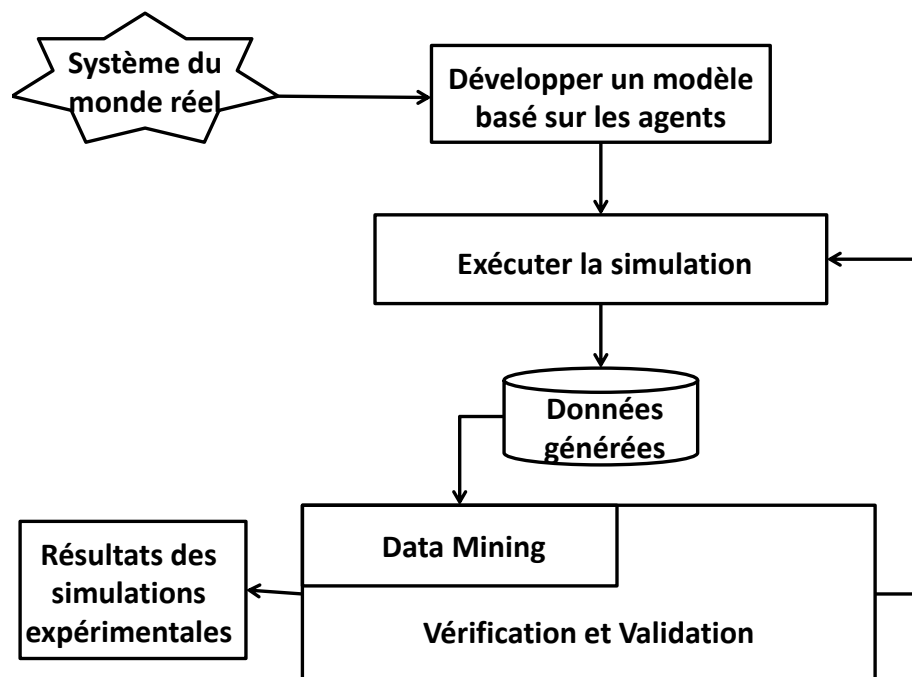


FIGURE 1.4 – Application du DM pour la vérification et la validation des simulations multi-agents [Baqueiro *et al.*, 2009]

1.5.1 Application exogène

L'application exogène est effective lorsque le Data Mining est utilisé sur les résultats finaux de la simulation afin d'analyser et de révéler des motifs pertinents permettant de mieux comprendre le comportement global du système. Le modèle des données de résultat créé permet alors de valider ou de réfuter les hypothèses initiales de la simulation. Ce rôle de validation est assuré par des outils tels que l'analyse de la variance, l'analyse des clusters ou des règles d'associations. Ce type d'application du Data Mining demeure assez confidentiel, en raison du nombre modeste des travaux qui s'y intéressent, et demande un travail de recherche plus poussé pour formaliser le processus de validation.

1.5.2 Application endogène

Des techniques d'intelligence artificielle sont employées dans la conception des agents afin d'adapter leur comportement intelligent. On citera en particulier l'apprentissage automatique et la modélisation de comportement dynamique. Ici le Data Mining est appliqué sur les résultats des exécutions précédentes de la simulation afin d'analyser et d'améliorer le comportement des agents, et ce en intervenant sur leurs paramètres initiaux [Remondino et Correndo, 2006] [Remondino et Correndo, 2005] d'où le terme « endogène ». Cette technique est similaire en quelque sorte au comportement d'un être humain dans une société où il apprend à partir de son expérience.

Ces précédents travaux ont validé le fait que le Data Mining permet de révéler des informations potentiellement significatives et utiles pour améliorer le comportement des agents. Manifestement, il a également le potentiel d'être l'origine de la description des comportements des agents. Dans les prochains chapitres de cette thèse, nous proposerons une nouvelle catégorie de combinaison entre Data Mining et simulation multi-agents. Les techniques de fouille de données y seront utilisées pour concevoir des agents à partir de l'observation du monde réel et des mesures prises dans l'environnement.

1.6 Conclusion

Dans ce chapitre, nous nous sommes intéressés aux problématiques concernant la modélisation et la construction des simulations multi-agents, et cela, en ciblant un cadre applicatif particulier qui est la simulation des espaces de vente.

De précédents travaux ont été élaborés traitant de simulation d'un tel système, aussi bien dans le domaine de la recherche que dans le domaine industriel. Afin de parvenir à réaliser de telles simulations, il est nécessaire de passer par une étape d'analyse de comportements de consommateurs. Cette étape est notablement fastidieuse et est traitée différemment d'une approche à une autre. Elle s'appuie généralement sur des études empiriques et par conséquent manque d'automatisme et d'objectivité.

Vu la complexité du système à simuler, une majeure partie des simulations proposées repose sur les systèmes multi-agents. Ces systèmes sont caractérisés par une notion originale : l'émergence qui permet de reproduire des phénomènes complexes dus aux interactions entre les agents tels que la carte d'occupation d'un espace de vente.

En ce qui concerne les méthodes de conception de simulation multi-agents, il existe des approches distinctes. Il est possible, notamment, de faire participer des humains, en tant qu'agents, dans la simulation multi-agents afin d'améliorer les protocoles d'interaction ; il est également possible de se focaliser uniquement sur les interactions des agents pour construire des comportements génériques et les séparer du traitement ; il est aussi possible d'intégrer des techniques de data mining dans les simulations multi-agents afin de formaliser des processus de vérification et de validation pour la simulation.

Cette intégration du Data Mining peut être faite de façon exogène sur le résultat final de la simulation afin de révéler des motifs permettant une meilleure compréhension du comportement global. Elle peut aussi être appliquée de façon endogène pour analyser et améliorer le comportement des agents, modifiant les résultats de simulation.

Les outils et les méthodes discutés tout au long de ce chapitre présentent certaines qualités dans des applications bien déterminées. Cependant, ils sont difficilement transposables sur d'autres types d'applications, notamment des simulations multi-agents reproduisant des comportements (de consommateurs dans un espace de vente) à faible interaction. De plus, ces approches sont majoritairement basées sur des études empiriques et nécessitent la présence d'un expert pour concevoir la simulation. Ceci rend la conception fastidieuse et subjective. Toutefois, l'application du Data Mining dans les simulations multi-agents a permis de faciliter cette phase de conception, mais uniquement dans une phase d'amélioration d'un système déjà existant.

L'environnement ciblé dans notre étude étant un espace de vente fréquenté par des clients aux profils différents, nous souhaitons définir une nouvelle méthodologie de conception agents, évitant autant que possible la collecte fastidieuse de données et l'intervention d'experts. Nous proposons ainsi une nouvelle approche pour l'intégration des techniques de Data Mining dans les simulations agents. Plus particulièrement, nous souhaitons automatiser, dans les limites du possible, la phase de conception des agents à partir de l'analyse de comportements de déplacement observés à partir d'une analyse vidéo du monde réel.

Dans le chapitre suivant, nous nous intéressons à certains travaux dans la littérature portant sur l'extraction de comportements de déplacement, à partir de l'observation. Nous nous focalisons sur l'extraction de comportements avec deux types d'apprentissage automatique, en l'occurrence la technique de l'Estimation Maximisation et le modèle de Markov caché.

Chapitre 2

Extraction de connaissances à partir de l'observation

Sommaire

2.1	Introduction	25
2.2	Présentation des besoins liés à l'observation du monde réel	26
2.3	Etat de l'art sur les techniques en traitement de trajectoires	28
2.4	Extraction de comportements par apprentissage de trajectoires	29
2.4.1	Réseaux Bayésiens Dynamiques Récurrents	30
2.4.2	Le Modèle de Markov Caché	32
2.4.3	Modèles dynamiques commutés	38
2.4.4	Modèle de mélange pour la classification automatique	41
2.5	Discussion et conclusion	48

2.1 Introduction

Nous nous sommes intéressés dans le premier chapitre aux simulations multi-agents et à leurs méthodes de conception mais sans se focaliser sur la manière dont les comportements d'agents sont extraits. Cependant, cette phase d'extraction des modèles d'agents est fondamentale. Elle représente la base sur laquelle le modèle de la simulation est construit.

La plupart des méthodologies de conception de simulation agents, dont celles que nous avons présentées dans le chapitre précédent, reposent sur l'exploitation des informations extraites d'enquêtes empiriques ou de connaissances d'experts du domaine.

Dans le cadre de cette thèse, nous cherchons à automatiser le plus possible cette phase initiale de conception en extrayant automatiquement des informations pertinentes sans avoir recours à des experts ou à des avis pouvant être subjectifs. Dans le contexte d'un espace de vente, il est ainsi possible d'utiliser les caméras de surveillance en tant que moyens d'observation des comportements réels des clients. A partir de ces observations, il est alors envisagé d'extraire, par modélisation, des comportements (simplifiés par rapport à la réalité), et cela afin de pouvoir ensuite les exploiter pour concevoir les agents de la simulation.

Pour cela, les techniques d'analyse de données sont capables d'extraire des connaissances pertinentes à partir d'un large volume de données issues de la mesure du monde réel. Par conséquent, ces techniques semblent adaptées pour extraire des informations synthétiques permettant ensuite d'être utilisées pour définir les comportements des agents.

Dans le cadre de cette thèse, nous nous intéressons plus précisément aux comportements de déplacement. Ce type de comportement est suffisamment riche pour réaliser plusieurs études expérimentales sur l'environnement étudié (description de l'affluence de certaines zones par exemple ou prédiction de l'affluence dans le cas de modifications apportées sur l'environnement).

Dans ce chapitre, nous présentons des méthodologies qui se rapportent à cette problématique d'extraction de connaissances. Elles consistent à extraire, à partir de données d'observation, les informations caractérisant les comportements d'individus que nous cherchons à reproduire par des agents dans la simulation.

En premier lieu, nous détaillons nos besoins en termes de types de données à traiter. Ensuite, nous exposons différents travaux traitant de l'apprentissage et l'extraction de connaissances dans un contexte de comportements d'individus observés, notamment les comportements de déplacement. Nous présentons particulièrement deux techniques d'apprentissage : l'algorithme EM (Espérance Maximisation) et les modèles de Markov cachés (MMC), qui permettent d'apprendre des comportements de façon totalement automatique et qui correspondent le plus à nos besoins.

2.2 Présentation des besoins liés à l'observation du monde réel

Dans cette section, la simulation d'un magasin de vente est retenue comme cadre applicatif. Dans ce contexte, nous décrivons ici nos besoins en termes de données à extraire, ainsi que leur type. Celles-ci seront utiles pour la méthodologie de conception de la simulation agents.

Les données qui pourront ainsi être exploitées peuvent provenir :

- des systèmes de vidéo-surveillance à partir desquels il est possible d'effectuer des suivis de clients, de détecter des mouvements spécifiques, notamment la gestuelle [Spiro *et al.*, 2010] (bouger une main, saisir un produit, le déposer,...), de détecter des émotions faciales [Pierce, 2012], ...
- des tickets de caisses contenant les listes de produits achetés pour chaque client, les dates et heures d'achat, permettant de déterminer les préférences des différents clients ainsi que leur logique d'achat,...
- des capteurs installés dans le magasin, notamment au niveau de l'entrée, afin de réaliser un comptage des personnes accédant à l'intérieur du bâtiment. Ces capteurs fournissent une information de flux de passage dans les zones équipées.

Ces supports hétérogènes aboutissent à des types de données aussi différents les uns que les autres. Etant donné que nous souhaitons aborder des comportements de déplacement, le moyen le plus efficace d'y parvenir est l'exploitation des enregistrements des caméras de surveillance. Partant du principe que la vidéo permet l'obtention de données suffisantes pour l'étude des déplacements des consommateurs à l'intérieur d'un établissement, nous nous limitons à ce support. En effet, à partir

des caméras de surveillance, il est possible de suivre les trajectoires des consommateurs, ce qui nous informe sur les parcours et temps de pause réalisés à l'intérieur du magasin et, par conséquent, sur l'intérêt que peuvent susciter certains rayons, voire certains produits, chez le consommateur, et cela indépendamment des achats réellement effectués.

Ainsi, à partir d'informations relatives aux déplacements, les paramètres des modèles d'agent peuvent être estimés, notamment en exploitant la connaissance des trajectoires et des régions spécifiques du magasin (figure 2.1). Ces dernières peuvent être représentatives des produits qui s'y trouvent (zone yaourt, zone beurre,...) ou grossièrement de la nature des rayons (rayon des produits frais, rayon des surgelés,...).

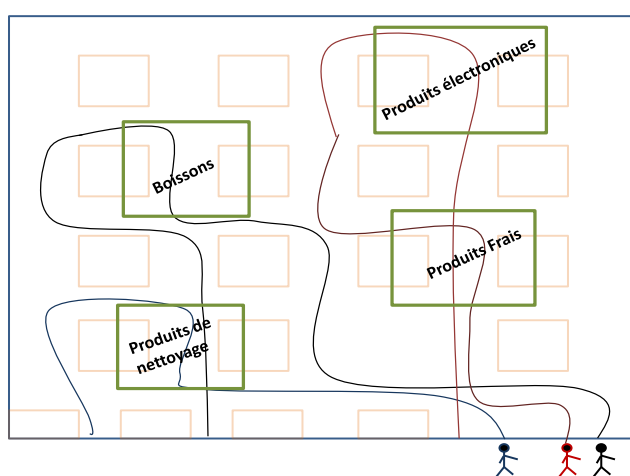


FIGURE 2.1 – Illustration du suivi de personnes dans un supermarché

L'environnement étant connu, il est possible de labelliser chaque région d'intérêt (zone accessible) selon les catégories de produits à proximité.

Ayant des connaissances sur l'environnement où évoluent les consommateurs, leurs trajectoires peuvent ainsi être interprétées en fonction des labels de chaque région d'intérêt visitée, de telle manière à obtenir une liste d'objectifs pour chaque individu qui pourra ensuite être exploitée pour la modélisation des agents correspondants.

Le temps passé dans chaque région peut aussi être mesuré. Selon l'arrêt de l'individu à un moment donné de la trajectoire, ou un passage furtif, le temps peut indiquer le niveau d'intérêt particulier pour une région spécifique.

Par ailleurs, ayant recueilli une quantité importante de données de déplacements observés concernant tous les individus impliqués dans l'environnement étudié, ces données pourront être exploitées en cherchant à isoler des motifs et des groupes correspondant à des observations récurrentes. Pour cela, il est nécessaire de recourir à des traitements d'analyse de données et d'extraction de connaissances. Ces traitements visent à extraire des comportements caractéristiques de déplacements pouvant être utilisés comme modèles pour les futurs agents de la simulation.

Il existe dans la littérature différentes sortes de techniques d'extraction de connaissances liées aux comportements d'individus à partir de trajectoires observées. Elles traitent aussi bien de l'évolution de trajectoires de piétons dans un environnement donné que de mouvements particuliers exercés par une ou plusieurs personnes. Un résumé de ces différents travaux est présenté dans les sections suivantes.

2.3 Etat de l'art sur les techniques en traitement de trajectoires

L'analyse de trajectoires humaines est associée à un vaste domaine de recherche qui a permis la mise en place d'un nombre important de solutions de vidéo-surveillance et de suivi de personnes. Leur principe général consiste à extraire, depuis des séquences vidéo, des régions d'intérêt où se trouvent des individus, de caractériser par des labels distincts chaque individu présent dans la scène, puis reconstruire le parcours du label en termes de coordonnées pixel dans la séquence. Les pionniers dans le domaine du suivi de personnes sont [O'Rourke et Badler, 1980], [Hogg, 1983] et [Siebel, 2003].

Ils ont depuis été suivis par une communauté importante de chercheurs et d'industriels. Les principaux verrous associés au suivi expliquent la diversité des méthodes rencontrées. Ainsi, le premier verrou concerne la capacité à extraire correctement de l'image les régions d'intérêt correspondant aux personnes à suivre. Trois approches sont ainsi rencontrées :

- celles basées sur une modélisation par régions homogènes [Gelgon, 1998] et [Siebel, 2003]. Les méthodes les plus répandues exploitent ainsi les probabilités d'appartenance des pixels à la personne en mouvement [Fieguth et Terzopoulos, 1997] et [Black et Jepson, 1998].
- celles qui utilisent les contours et les modèles 2D et 3D associés décrivant l'enveloppe d'une personne [Rigoll *et al.*, 2000]. En général, les modèles 2D sont les plus appropriés car plus simples [Ren *et al.*, 2005], [Hogg, 1983], la 3D étant réservée aux situations de suivi plus complexes, notamment au suivi des mouvement des membres [Gavrila et Davis, 1996].
- les approches directes qui exploitent un filtrage directement appliqué à l'image afin de rehausser les zones de mouvement apparent des pixels, chaque mouvement étant perçu comme une variation de niveaux lumineux [Masoud et Papanikolopoulos, 2003] et [Bretzner *et al.*, 2002].

Le second verrou important concerne la capacité à suivre les trajectoires de façon robuste, notamment en considérant les problèmes d'occultation de la scène et de l'individu, le suivi en parallèle d'un grand nombre de cibles, le problème d'ouverture où les déplacements, selon la normale du plan caméra, sont plus difficilement détectables,... La plupart des techniques développées en réponse à ces difficultés sont inspirées des techniques historiques de suivi de cibles dans le traitement multi-antenne des signaux radar. Deux principales catégories ont été étudiées :

- les techniques déterministes. Elles sont basées sur des techniques de filtrage linéaire [Blackman, 1986] [Bar-Shalom et Fortmann, 1988] ou de filtrage de Kalman [Rerkrai et Fillbr, 2004]. Bien que simples à exploiter, vu leurs hypothèses de modélisation (continuité des signaux, bruit gaussien,...), elles restent cependant limitées par un fort risque de perdre le suivi.

- les techniques probabilistes. Elles consistent à estimer la trajectoire à partir d'une fonction de vraisemblance qui modélise l'association observations / trajectoire dont la technique la plus ancienne est celle de [Buechler et Smith, 1975]. Dans les dernières années, ces techniques ont été améliorées par l'introduction des graphes et des réseaux bayésiens [Rota, 1998], [Brand et Kettner, 2000] et [Buxton et Gong, 1995]. Les meilleures performances ainsi obtenues sont notamment dues à la capacité de ces techniques à reconstruire une trajectoire en présence d'occultations en permettant le regroupement des segments de trajectoire appartenant au même individu.

Enfin, parmi les principaux verrous scientifiques associés au suivi de personne, le dernier listé ici consiste en l'analyse de la trajectoire et la reconnaissance de l'action faite par la personne suivie. En matière de reconnaissance d'actions ou de comportements, l'analyse des mouvements est traitée selon deux principes. Soit la séquence vidéo entière est utilisée pour caractériser un mouvement, soit le mouvement est défini par itération, image après image [Lipton *et al.*, 1998] [Zhao. et Huang, 2002] [Khan et Shah, 2003].

Le premier principe reste le plus répandu car il permet une analyse plus fouillée des actions et des activités présentes dans la séquence vidéo. Sous ce principe est regroupé un vaste ensemble de méthodes basées sur l'analyse de primitives extraites de l'image, puis sur l'emploi de techniques d'apprentissage. Considérant les primitives utilisées, elles peuvent être soit locales [Savarese *et al.*, 2008], [Md Sabri *et al.*, 2012] soit globales [Gorelick *et al.*, 2007]. Le premier type de primitives est plus robuste car la plupart d'entre elles sont invariantes aux changements de point de vue, à l'apparence de la personne ainsi qu'aux occlusions partielles.

Cependant, bien qu'améliorée par l'usage de primitives plus robustes, la reconnaissance ou classification d'actions reste confrontée au fait que ces informations peuvent être incomplètes et plus ou moins variantes en termes d'échelles de taille et de temps. Cela nécessite le recours à des techniques d'apprentissage supervisé et non-supervisé spécifiques. A partir d'un ensemble de primitives robustes et d'un classifieur performant, il est alors possible de modéliser les principaux motifs de parcours et les principaux comportements des individus dans la scène (par exemple, sens de déplacement des clients, point de passage régulier dans une zone délimitée, séquence de mouvement marche -arrêt-marche correspondant au passage d'un obstacle,...).

De nombreuses techniques d'apprentissage ont ainsi été appliquées : les MMC [Yamato *et al.*, 1992] [Han *et al.*, 2007], les mélanges de modèles gaussiens [Stauffer et Grimson, 1999] [Thome, 2007], les techniques SVM [Ronfard *et al.*, 2002], [Niu *et al.*, 2003],... Parmi ces techniques et compte tenu de notre cadre applicatif, les techniques basées sur les mélanges gaussiens ainsi que les MMC ont retenu notre attention.

2.4 Extraction de comportements par apprentissage de trajectoires

Dans cette section, nous portons notre intérêt sur des travaux traitant de l'identification et l'extraction de comportements à partir de séquences vidéo. Particulièrement, nous abordons des études sur la détermination et l'analyse des activités humaines se basant sur leurs trajectoires, notamment

dans le domaine de surveillance, dans certaines opérations de marketing sur des espaces de vente ou des opérations à des fins statistiques dans un environnement donné,...

De nos jours, de nombreux endroits publics sont dotés de caméras de surveillance (les espaces commerciaux, les stations de métro, certaines rues ou avenues,...). Les vidéos sont généralement visionnées par des responsables de la sécurité ou des spécialistes qui doivent parfois surveiller plusieurs écrans en même temps. Dans ce contexte, il est possible que des événements ou informations utiles ou graves à notifier ne soient pas vus.

Afin d'assister les opérateurs de télésurveillance ou simplement réaliser des études de comportements de déplacement, des travaux de recherche ont été menés dans le but d'améliorer les systèmes de vidéo surveillance notamment en leur permettant d'identifier automatiquement des activités. Il s'agit, en l'occurrence, d'inspecter les actions des individus en déterminant leur nature et de distinguer les comportements inappropriés le cas échéant.

Les travaux que nous présentons, ci-après, traitent de cette problématique. Néanmoins, chacun de ces travaux procède différemment s'appuyant sur des techniques distinctes. Un grand nombre d'entre elles s'appuie sur une représentation d'activité structurée en machine à états où chaque état représente un comportement élémentaire. Parcourir la structure permet, en l'occurrence, de retrouver un comportement de déplacement complet. Outre cette représentation à base de graphes, d'autres études exploitent des représentations de trajectoires complètes d'emblée à partir d'algorithmes de classification. Nous abordons ces algorithmes dans la dernière partie de la section suivante.

2.4.1 Réseaux Bayésiens Dynamiques Récurrents

Des travaux se sont penchés sur une forme particulière des Réseaux Bayésiens Dynamiques, conçus et appris à partir de données extraites de séquences vidéo, pour la reconnaissance de comportements humains. L'originalité de cette approche réside dans l'utilisation de réseaux bayésiens récurrents (RBR) [Moënné-Loccoz *et al.*, 2003] afin de modéliser l'évolution temporelle du comportement et interpréter des événements se produisant dans une scène particulière.

Pour ce faire, un ensemble de descriptions et de connaissances *a priori* sont prises en compte, notamment :

- **l'état** de l'objet mobile à des intervalles de temps donnés (immobile / marche / court) ;
- **l'évènement** qui dénote un changement d'état (s'arrêter / commencer à marcher) ;
- **le scénario** qui combine un ensemble d'états ou d'évènements (s'arrêter devant un article dans un supermarché) ;
- les connaissances sur **l'environnement de la scène** (linéaire, produits, portes, murs,...) ;
- les connaissances de **l'expert** qui représente et modélise les états, les événements et les scénarios.

A partir des traitements faits sur l'image, de la description et des connaissances d'une scène ciblée, les RBR permettent de reconnaître de façon hiérarchisée des comportements humains allant des états et des événements jusqu'aux scénarios. Pour illustrer le fonctionnement des RBR, la figure 2.2 représente la structure d'un comportement d'agitation globale représentant un acte de violence

dans un environnement donné. Le RBR, dans ce cas de figure, prend comme données d'entrée les valeurs des fonctions visuelles (correspondant à des mesures estimées à partir des images (voir section 2.3)) durant une période définie (l'évolution de la taille de la foule (F1), l'accélération de la foule (F2) et l'évolution de l'orientation de la foule (F3)). Le réseau mémorise également l'information de l'occurrence précédente du comportement par dépendance récurrente (de l'agitation externe à $t - 1$ vers l'agitation externe à t). Dans la figure donnée, nous remarquons deux structures de RBR décrivant la dépendance temporelle de deux comportements d'agitation d'un groupe de personnes : interne (B1) et externe (B2). L'agitation globale (B), ou l'occurrence d'un acte de violence, est ensuite inférée par classification bayésienne.

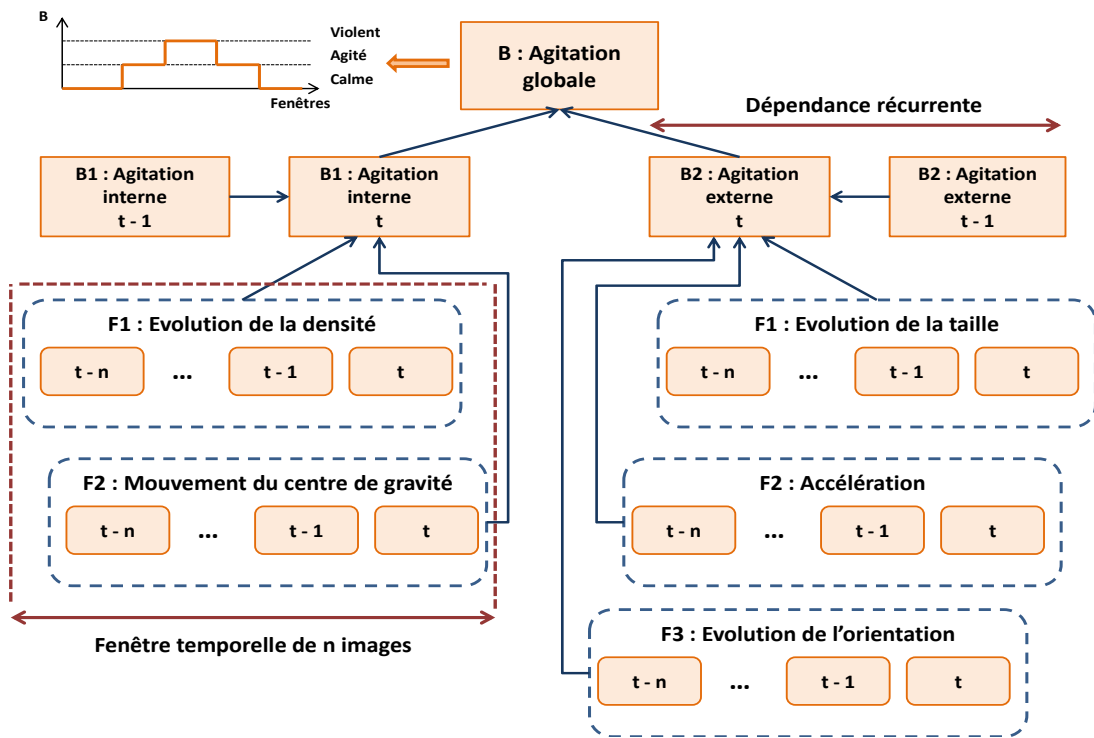


FIGURE 2.2 – Réseau Bayésien Récurrent pour la reconnaissance de niveau d'agitation globale d'un groupe de personnes [Moënné-Loccoz *et al.*, 2003]

Formellement, B étant le comportement à reconnaître, on dispose d'un ensemble de dépendances $O = \{R, F_{1t}, F_{1t-1}, \dots, F_{1t-n}, F_{2t}, F_{2t-1}, \dots, F_{2t-n}, \dots, F_{kt}, F_{kt-1}, \dots, F_{kt-n}\}$ avec R : la dépendance récurrente, k : le nombre de caractéristiques visuelles (par exemple : évolution de l'orientation, accélération), $n + 1$: la taille de la fenêtre temporelle et F_{it} : la dépendance à la caractéristique visuelle i à l'instant t . Pour simplifier le formalisme de cet ensemble, il est préférable de le présenter par $O = \{O_1, O_2, \dots, O_{nk+1}\}$ où O_1 correspond à R , O_2 correspond à F_{1t} .

Pour évaluer l'occurrence du comportement B par rapport au comportement opposé ($\bar{B}$), les auteurs utilisent une comparaison des deux probabilités conditionnelles $P(B|O)$ et $P(\bar{B}|O)$, comme suit :

$$\frac{P(B|O)}{P(\bar{B}|O)} = \frac{\prod_i P(O_i|B)}{\prod_i P(O_i|\bar{B})} \quad (2.1)$$

Où $P(O_i|B)$ est calculée dans la phase d'apprentissage à partir d'un ensemble de couples $(b, o)_i$ manuellement labellisés, représentant la valeur du comportement (b) et la valeur de toutes les dépendances (o). Il s'agit de décider pour chaque image, ayant ses caractéristiques visuelles et ayant l'information de la période précédente, si le comportement d'intérêt se réalise ou pas.

L'estimation des paramètres du réseau (ie les probabilités conditionnelles) dans la phase d'apprentissage repose sur le choix d'un certain nombre d'exemples de flux vidéo, dont 80% décrivent le comportement qu'on cherche à modéliser et 20% décrivent le comportement opposé. Tous les exemples sont annotés par l'expert. Afin de reconnaître un comportement à partir de nouvelles séquences vidéo, le rapport $\frac{P(B|O)}{P(\bar{B}|O)}$ est calculé, indiquant ainsi le degré d'appartenance du comportement courant au comportement appris.

Cette approche est intéressante en termes de capacités à capturer un comportement quelle que soit son échelle de temps. Elle permet également de limiter les difficultés de reconnaissance face aux incertitudes, aux manques de connaissance initiale, et plus particulièrement, face à la dépendance avec l'environnement. Une évolution temporelle d'une agitation de groupe, modélisé comme le montre la figure 2.2, a été correctement détectée. En effet, un comportement de violence est constaté pour chaque exemple positif et un comportement de non-violence est constaté pour deux faux exemples.

Cependant, la phase d'annotation des séquences vidéo (pour l'apprentissage des probabilités conditionnelles) dépend fortement de l'expert humain. De plus, il s'agit d'une tâche fastidieuse qui risque d'être très lourde dans le cas où plusieurs comportements doivent être traités avec un volume de données très important.

2.4.2 Le Modèle de Markov Caché

Dans l'état actuel des connaissances, le modèle de Markov caché (Hidden Markov Model ou MMC) représente l'un des outils de modélisation les plus utilisés dans le traitement de la parole et les problèmes de reconnaissance associés [Rabiner et Juang, 1986]. Il apporte également des résultats prometteurs dans le domaine de reconnaissance des activités [Nguyen et Venkatesh, 2005].

D'une manière générale, un MMC est un modèle statistique permettant de représenter des systèmes modélisables sous la forme d'un processus de Markov¹ dont les états sont cachés [Rabiner et Juang, 1986]. Le rôle d'un MMC est de construire un modèle qui peut caractériser la présence de symboles observés. Le modèle obtenu peut être utilisé par la suite, pour identifier, reconnaître ou générer d'autres séquences d'observation. Les paramètres d'un tel modèle sont définis comme suit :

- N : le nombre d'états dans le modèle ;
- M : le nombre d'observations ;
- $V = \{v_1, v_2, \dots, v_M\}$: un ensemble discret des symboles d'observations possibles ;

1. un processus stochastique dont l'état courant ne dépend pas du passé

- $A = \{a_{ij}\}, a_{ij} = P(q_j \text{ à } t+1 | q_i \text{ à } t)$ avec $1 \leq i, j \leq n$: la distribution de probabilité des transitions entre les états q_i et q_j ;
- $B = \{b_j(k)\}, b_j(k) = P(v_k \text{ à } t | q_j \text{ à } t)$: la distribution de probabilité des symboles d'observation pour chaque état q_j ;
- $\pi = \{\pi_i\}, \pi_i = P(q_i \text{ à } t=1)$: la distribution initiale des états.

Dans le domaine de la reconnaissance vocale, de nombreux travaux utilisent les signaux vocaux de mots de vocabulaire, prononcés par plusieurs personnes, et les divisent en séquences de symboles (répertoire de symboles) en utilisant des techniques de quantification vectorielle [Rabiner et Juang, 1986]. A partir de ces symboles, chaque mot est représenté par un MMC avec n états. La reconnaissance est alors effectuée par l'évaluation des MMC obtenus étant donné une séquence d'observation et en sélectionnant le score le plus élevé des modèles. Étant donné que les signaux vocaux sont généralement représentés par des courbes, ils peuvent être assimilés à des trajectoires (dans l'intention d'apprendre des comportements de déplacement). De façon analogue, ces trajectoires peuvent être divisées en séquences de mouvements élémentaires. Il peut s'en suivre la création de plusieurs MMC représentant des ensembles de trajectoires équivalentes. De cette manière, il est possible de reconnaître ou d'identifier certaines nouvelles trajectoires sur cette base de modèles.

Dans la littérature, les MMC sont également exploités afin d'extraire certaines informations importantes, répondant à certains problèmes d'exploitation automatique de données de vidéo-surveillance. Pour ce faire, diverses méthodes exploitent les avantages des MMC pour détecter et reconnaître des trajectoires à partir de contenus vidéo. Dans [Mlich et Chmelar, 2008], les auteurs ont expérimenté l'application de MMC sur des trajectoires de personnes dans une station souterraine. En premier lieu, ils ont défini les classes de toutes les trajectoires normales (passer de gauche à droite en arrière plan de la scène, passer de droite à gauche, passer la première barrière de péage en venant de la partie droite de la scène,...). Ensuite, un modèle initial est déterminé pour chaque classe de trajectoire. Enfin, les trajectoires des séquences vidéo de la même scène sont annotées manuellement afin d'adapter les modèles correspondants. L'idée est qu'à partir d'une nouvelle trajectoire on est capable de reconnaître à quelle classe elle appartient. Dans le cas où le degré de correspondance est faible avec toutes les classes, la trajectoire est considérée comme anormale. Il s'agirait donc d'un événement inhabituel méritant une attention particulière. La figure 2.3 récapitule ce processus.

[Hervieu *et al.*, 2007] proposent une méthode permettant d'extraire des classes contenant des mouvements similaires en termes de formes de trajectoires intrinsèques et vitesses d'objets sans exploiter l'information explicite de la structure de la scène. Ils appliquent une méthode de reconnaissance basée sur les MMC pour capturer l'évolution temporelle de l'évènement dynamique considéré. Pour ce faire, des descripteurs sont calculés pour chaque trajectoire T_k définie par un ensemble de positions successives d'un objet suivi dans une séquence d'images ($T_k = \{(x_1, y_1), \dots, (x_{n_k}, y_{n_k})\}$). Ces descripteurs permettent de considérer uniquement l'aspect général des trajectoires. En d'autres termes, il s'agit d'obtenir une représentation invariante à la translation, au facteur d'échelle ainsi qu'à la rotation. Un vecteur de descripteurs est alors obtenu pour chaque trajectoire, défini par $\phi = [\dot{\gamma}_1, \dot{\gamma}_2, \dots, \dot{\gamma}_{n_k}]$, où chaque point est représenté par la courbure locale et la norme de la vitesse locale représentées par $\dot{\gamma}_t$.

Dans le but d'obtenir une représentation invariante des trajectoires et considérer la causalité temporelle qui leur est inhérente, les auteurs ont recours aux MMC.

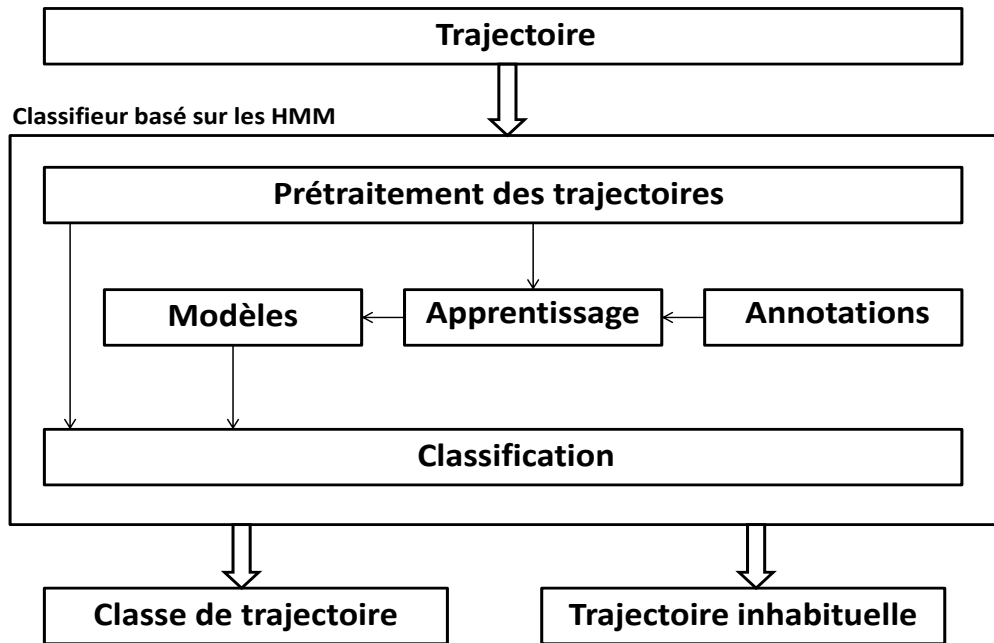


FIGURE 2.3 – Traitement de trajectoires [Mlich et Chmelar, 2008]

Chaque trajectoire est, donc, modélisée par un MMC. Les états n'étant pas connus *a priori*, ils sont définis par une quantification effectuée sur les descripteurs $\dot{\gamma}$ afin de maintenir les MMC à un nombre d'états limité et représentatif. Pour ce faire, un intervalle $[B_1, B_2]$ est fixé retenant un pourcentage P , également fixé, de toutes les valeurs de $\dot{\gamma}$ mesurées autour de leur moyenne. L'intervalle $[B_1, B_2]$ est ensuite fragmenté en N' segments correspondants à des états du MMC. Les intervalles $[-\inf, B_1]$ et $[B_2, +\inf]$ représentent les états extrémaux. Par conséquent, chaque trajectoire est modélisée par un MMC à $N = N' + 2$ états avec les observations V représentant les $\dot{\gamma}_t$ calculés (voir figure 2.4).

Pour estimer le modèle de Markov caché, de nombreux travaux se basent sur l'algorithme 1 de Baum-Welch. L'algorithme de Baum-Welch (généralisation de l'algorithme d'estimation-maximisation présenté dans la section 2.4.4.2) est appliqué pour estimer ses paramètres à partir d'une séquence d'observation et d'une initialisation aléatoire du MMC.

Dans le cas de la modélisation d'une trajectoire quantifiée en vecteur de descripteurs, les probabilités conditionnelles d'observation $b_i(v_t)$ (avec v_t l'observation au temps t) sont modélisées selon des gaussiennes centrées en μ_i (μ_i représente le centre du fragment correspondant à l'état i). L'écart type σ de ces gaussiennes correspond à la demie largeur du fragment. A et π sont estimés à partir de la méthode par moindres carrés portant sur un processus de comptage [Ford et Moore, 1998], tel que pour une trajectoire T_k comprenant n_k observations :

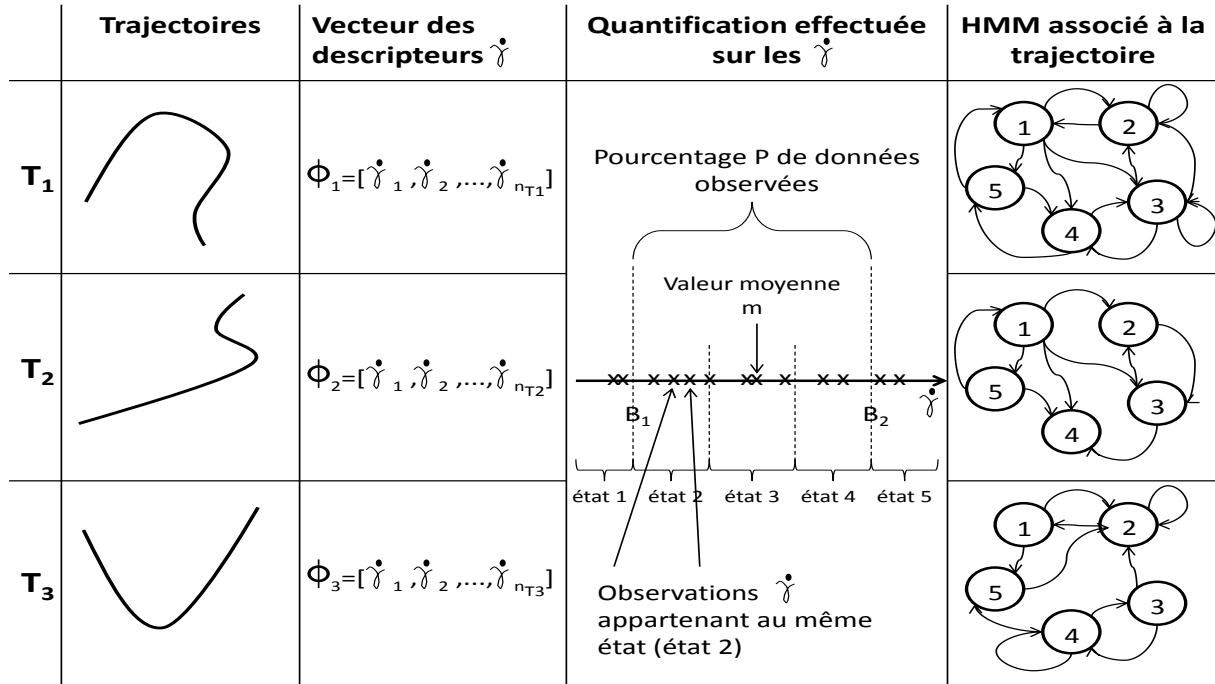


FIGURE 2.4 – Modélisation de trajectoires par MMC

$$a_{ij} = \frac{\sum_{n=1}^{n_k-1} H_n^{(i)} H_{n+1}^{(j)}}{\sum_{n=1}^{n_k-1} H_n^{(i)}} \quad \text{et} \quad \pi_i = \frac{\sum_{n=1}^{n_k} H_n^i}{n_k} \quad (2.2)$$

Avec $H_t^{(i)} = P(\dot{\gamma}_t | q_t = i)$ correspond à un poids pour le processus de comptage effectué.

Dans les travaux de Hervieu [Hervieu *et al.*, 2007], afin d'évaluer les performances de classification et de reconnaissance utilisant les MMC des trajectoires, des classes de trajectoires ont été extraites de vidéos de Formule 1 et de vidéos de compétition de ski. Chaque classe est composée de trajectoires extraites de plans filmés par une même caméra. Une classe peut ainsi englober différentes classes représentant un type de trajectoire en particulier telles que la "descente" ou le "slalom" en ski ; d'où l'invariance à la représentation des trajectoires de ces classes. La reconnaissance est réalisée au moyen d'une mesure de similarité (distance de Rabiner) entre deux MMC associés à deux trajectoires. Les résultats en pourcentage de bonne classification semblent très satisfaisants à comparer à d'autres méthodes. Néanmoins, la classification dans ce cas de figure repose sur une base de classes ou agglomération de classes existante et préalablement sélectionnées, grâce à la connaissance des champs des caméras ou les types de figures filmées.

Porikli [Porikli, 2004] fournit des fonctionnalités de suivi améliorées favorisant la détection des schémas des trajectoires communes. Il introduit des représentations de l'espace des paramètres des objets suivis. Cet espace de paramètres est ensuite fractionné en utilisant des variables MMC continues ayant une structure fixe. Afin d'ajuster les paramètres du MMC pour chaque trajectoire, un

Algorithm 1 L'algorithme de Baum-Welch

Entrées : $O = (o_1, o_2, \dots, o_T)$: une séquence d'observations

N = le nombre d'états cachés

1- Initialiser les paramètres A, B et π (aléatoirement par exemple) du modèle λ

2- calculer les probabilités selon l'algorithme "Forward-Backward" :

- Forward : la probabilité d'être à l'état i , à l'instant t , ayant eu les observations de o_1 à o_t :

$$\alpha_t(i) = P(q_t = i, o_1, \dots, o_t | \lambda)$$

- Backward : la probabilité d'avoir $o_{t+1}, \dots, o_T$ comme observations ultérieures à l'instant t , étant à l'état i :

$$\beta_t(i) = P(o_{t+1}, \dots, o_T | q_t = i, \lambda)$$

3- Utiliser les probabilités de Forward-Backward pour estimer deux matrices Γ et Ξ telles que :

- Ξ est une matrice dont les coefficients $\xi(i, j)$ représentent la probabilité d'être dans l'état i à l'instant t et de passer dans l'état j à l'instant $t+1$:

$$\xi(i, j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)}$$

- Γ est une matrice dont les coefficients $\gamma_t(i)$ représentent la probabilité d'être dans l'état i à l'instant t , sachant l'observation O et le modèle λ :

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) = \frac{\alpha_t(i) \beta_t(i)}{P(O | \lambda)}$$

4- Réestimer les paramètres du modèle :

$$\pi_i = \gamma_1(i) \quad 1 \leq i \leq N$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad 1 \leq i \quad \text{et} \quad j \leq N$$

$$b_i(k) = \frac{\sum_{t=1}^T 1_{o_t=o_k} \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)}$$

5- Répéter 2-3-4 jusqu'à ce que les paramètres convergent.

post-traitement basé sur la catégorisation est entrepris pour obtenir des schémas de trajectoires. Dans [Bashir *et al.*, 2006] et [Bashir *et al.*, 2007], les auteurs segmentent un ensemble de trajectoires d'apprentissage appartenant à une classe en sous-trajectoires pour ensuite construire un MMC avec un nombre d'états égal au nombre de sous-trajectoires.

Une autre étude, menée par les auteurs de [Nguyen et Venkatesh, 2005] s'appuie sur les MMC hiérarchiques afin de découvrir des structures d'activités complexes. Dans cette étude, l'environnement est discrétisé en cellules qui sont ensuite utilisées comme observations des MMC. Les auteurs sont capables d'apprendre des comportements de haut niveau se basant sur des comportements de bas niveau ou primitifs décrits par des trajectoires. Afin d'illustrer cette approche, prenons un des exemples introduits par les auteurs. "prendre un café" représente un comportement complexe. Etant dans une salle équipée d'une porte, une machine à café et une table, les comportements de bas niveau constituant, dans un ordre chronologique donné, le comportement "prendre un café" pourraient être définis en trois composantes trajectoires, notamment "de la porte à la machine à café", ensuite "de la machine à café à la table" et enfin "de la table à la porte". Dans cette approche, le comportement de haut niveau est représenté initialement par un graphe complet de tous les comportements primitifs. L'apprentissage consiste à élaguer les arcs aux faibles poids, tout en gardant l'ordre chronologique des comportements primitifs sous forme de réseau bayésien dynamique. La figure 2.5 décrit une représentation de ce modèle.

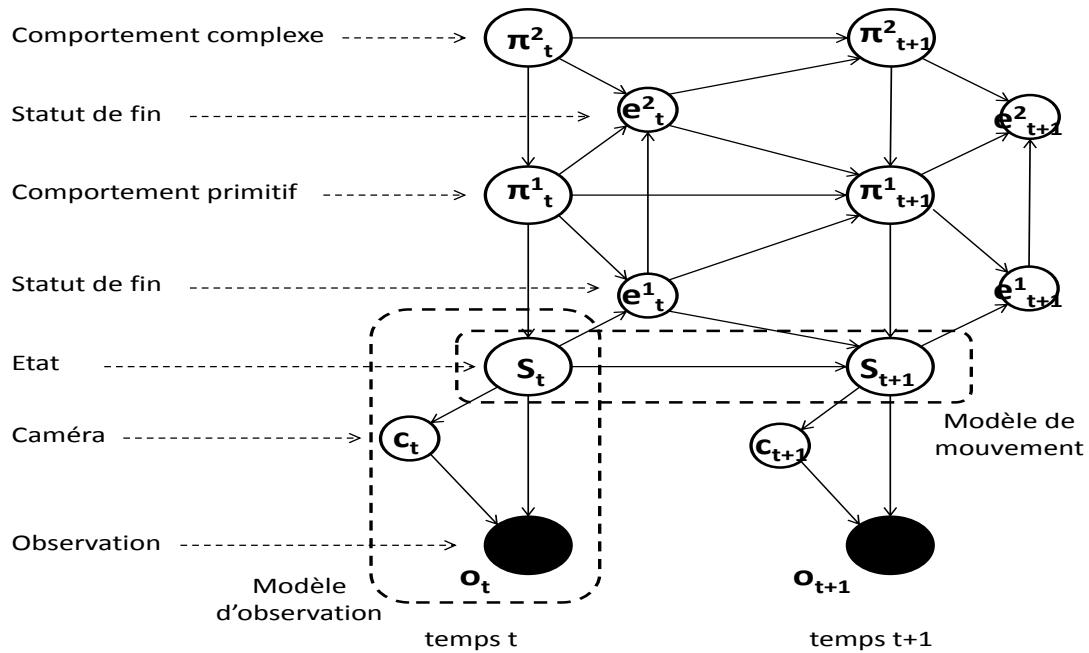


FIGURE 2.5 – Représentation en réseau bayésien dynamique du modèle de comportement sur deux laps de temps [Nguyen et Venkatesh, 2005]

Apprendre les probabilités des liens à partir des données d'apprentissage, revient à estimer les paramètres de π^k (tel que k correspond à un niveau de hiérarchie). Il s'agit donc d'estimer les paramètres suivant :

- $I^{\pi^2}(\pi^1)$: la probabilité que π^2 sélectionne le comportement primitif π^1 pour démarrer son exécution.
- $A^{\pi^2}(\pi^1, \pi^{1'})$: la probabilité que π^2 sélectionne le prochain comportement primitif $\pi^{1'}$ à exécuter à l'achèvement du comportement primitif courant π^1 .
- $E^{\pi^2}(\pi^1)$: la probabilité que π^2 s'achève lorsque le comportement primitif courant π^1 se ter-

mine.

Une expérimentation de reconnaissance de comportements a été réalisée sur 30 séquences de tests utilisant les comportements complexes appris. Le taux de précision qui se rapporte à la performance de reconnaissance au niveau des comportements complexes s'élève à 100%. Toutefois, dû à la complexité des comportements ainsi modélisés, il reste fastidieux d'appliquer cette méthode sur un environnement plus complexe tel qu'un magasin de vente par exemple. En effet, il apparaît difficile dans ce cas de reconnaître et annoter les comportements complexes à partir de tous les comportements primitifs (nombreux et permettant des combinaisons innombrables).

Dans le cadre d'un apprentissage de comportements de déplacement, supposons que nous disposons de séquences d'observation, représentées par des listes de position dans un environnement donné, décrivant ces comportements. Si le MMC est appris suivant des séquences de déplacement similaires (donc choisies et annotées), l'apprentissage serait considéré comme supervisé, car nous connaissons la catégorie du déplacement. Dans ce cas l'utilisation du MMC est semblable aux approches exposées auparavant dans cette section. Un ensemble de MMC appris sur différents types de déplacement pourrait être utilisé pour reconnaître le comportement de déplacement. En revanche, si un unique MMC est appris à partir de différentes séquences de déplacement correspondant à différents types, l'apprentissage pourrait être donc considéré comme non supervisé. Dans ce cas, il serait intéressant d'étudier si ce modèle MMC peut être utilisé en tant que structure générant des séquences d'observations reproduisant différentes trajectoires apprises.

2.4.3 Modèles dynamiques commutés

Certaines méthodes de reconnaissance d'activités à partir de séquences vidéo portent sur l'étude des trajectoires des piétons à partir de modèles dynamiques commutés [Nascimento *et al.*, 2010]. Dans le cas d'un environnement représentant l'entrée d'un magasin dans un centre commercial, les auteurs dans [Nascimento *et al.*, 2005] retiennent quatre classes d'activités communes suivant quatre itinéraires typiques ("entrer", "sortir", "passer" et "chercher"). Ces itinéraires décrivent des trajectoires pouvant être segmentées en cinq actions élémentaires ("se déplacer vers le haut", "se déplacer vers le bas", "se déplacer vers la gauche", "se déplacer vers la droite" et "s'arrêter"). Des modèles dynamiques commutés de ces différents types de déplacement viennent décrire les différentes trajectoires possibles.

La figure 2.6 représente les quatre classes d'activité. Pour chacune d'entre elles, la figure présente les différentes étapes permettant d'aboutir à un modèle dynamique commuté à partir des trajectoires extraites des séquences vidéos. Plus précisément, les données d'entrée de cette méthode consistent en un ensemble de trajectoires observées décrivant les positions successives des centres de gravité des personnes observées en vidéo. Chaque trajectoire de longueur n est donc traduite en une variable $x = (x_1, \dots, x_n)$ avec $x_t \in \mathbb{R}^2$.

Les trajectoires étant non stationnaires, il est difficile de les décrire avec un unique modèle dynamique. Ce qui explique l'emploi de M modèles dynamiques commutés déterminant différentes directions de mouvement. Dans l'exemple ci-dessus $M = 5$ (pour droite (k1), haut (k2), bas (k3), arrêt (k4), gauche (k5)). x_t est, par conséquent, modélisé par l'équation du système dynamique commuté suivante :

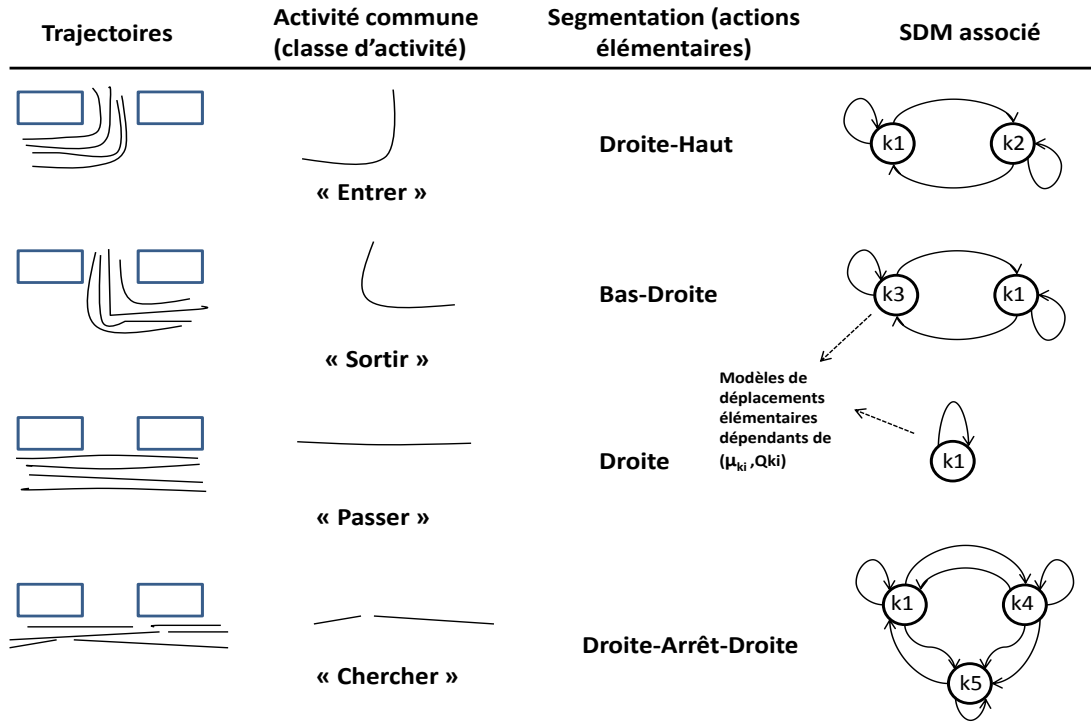


FIGURE 2.6 – Mécanisme de modélisation de trajectoire basé sur les modèles dynamiques commutés

$$x_t = x_{t-1} + \mu_{k_t} + Q_{k_t}^{1/2} w_t \quad (2.3)$$

où

- $k_t \in 1, \dots, M$ le label du modèle actif à l'instant t .
- $\{\mu_1, \dots, \mu_M\}$ les moyennes des vecteurs de déplacement pour chaque mouvement.
- $\{Q_1, \dots, Q_M\}$ les matrices d'incertitude associées à chaque mouvement élémentaire.
- w_t des échantillons indépendants d'un vecteur aléatoire gaussien de moyenne nulle et de covariance identité.

L'équation (2.3) peut s'écrire comme suit :

$$x_t - x_{t-1} = \Delta x_t = \mu_{k_t} + Q_{k_t}^{1/2} w_t \quad (2.4)$$

L'ensemble $\{\Delta x_t; t \in \mathbb{N}\}$ représente donc les observations. Par conséquent, et selon k_t , cette équation décrit un processus incrémental indépendant comme on peut le voir dans la figure 2.7.

Pour tout modèle d'activité m , des trajectoires d'apprentissage sont collectées. On obtient alors un vecteur $\Delta X^m = \{\Delta x_1^m, \Delta x_2^m, \dots, \Delta x_N^m\}$ contenant tous les déplacements du $m^{\text{ème}}$ modèle d'activité ayant, respectivement, comme estimations standards de la moyenne et de la matrice de covariance :

$$\hat{\mu}_m = \frac{1}{|\Delta X^m|} \sum \Delta X_t^m, \quad \hat{Q}_m = \frac{1}{|\Delta X^m|} \sum (\Delta X^m - \hat{\mu}_m)^t (\Delta X^m - \hat{\mu}_m)$$

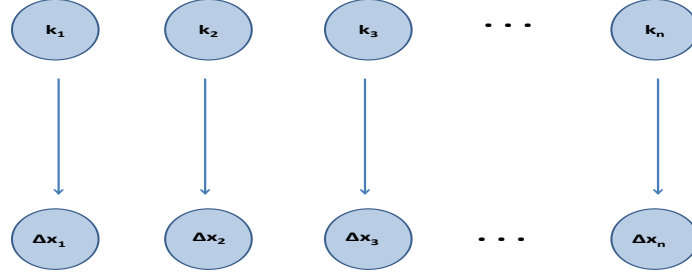


FIGURE 2.7 – Architecture du processus incrémental indépendant basé sur les modèles dynamiques commutés [Nascimento *et al.*, 2005]

Les modèles d'activité et les paramètres correspondants étant appris, il reste à classifier des trajectoires tests. Pour cela, on considère une trajectoire échantillon $x = (x_1, \dots, x_n)$ décrite par T segments correspondants à des actions élémentaires, et on calcule le maximum de sa log-vraisemblance dans l'espace des modèles. Cette dernière est définie comme suit :

$$\begin{aligned}
 L(\Delta x_1, \dots, \Delta x_n | k_1, \dots, k_T, t_1, \dots, t_{T-1}) &= \sum_{j=1}^T \sum_{i=t_{j-1}}^{t_j} \log p(\Delta x_i | k_j) \\
 &= \sum_{j=1}^T \sum_{i=t_{j-1}}^{t_j} \log \mathcal{N}(\Delta x_i | \mu_{k_j}, Q_{k_j}) \quad (2.5)
 \end{aligned}$$

Où :

- $k_1, \dots, k_T$ est la séquence des labels du modèle décrivant la trajectoire.
- $t_i, i = 1, \dots, T - 1$ est l'instant où une commutation d'un modèle élémentaire k_i à un autre k_{i+1} se produit.

A un niveau plus élevé, concernant les classes d'activités ("Entrer", "Sortir"...), la séquence de labels $k = (k_1, \dots, k_n) \in \{1, \dots, M\}^n$ d'une trajectoire appartenant à une classe d'activité a , considérée comme cachée à cause de sa nature non observable (contrairement à la séquence $x = (x_1, \dots, x_n)$ obtenue de la vidéo), représente un échantillon d'une chaîne de Markov cachée avec B_a sa matrice de transition de dimension $M \times M$ spécifique à chaque classe. On obtient par conséquent :

$$p(k|a) = p(k|B_a) = p(k_1|a) \prod_{t=2}^n B_{k_{t-1}, k_t}^{(a)} \quad (2.6)$$

Le classifieur, étant basé sur des modèles générateurs de trajectoires conditionnels à une classe [Nascimento *et al.*, 2010], est obtenu à partir de :

$$p(x|a) = \sum_{k \in \{1, \dots, M\}^n} p(x, k|a) = \sum_{k \in \{1, \dots, M\}^n} p(x|k) p(k|a) \quad (2.7)$$

A partir des données d'apprentissage, les paramètres des modèles ainsi que les matrices de transitions sont estimés en utilisant une version de l'algorithme Baum-Welch représentant une méthode généralisée de l'algorithme EM [Nascimento *et al.*, 2007].

La finalité de ce travail est d'utiliser les modèles obtenus pour la classification et la reconnaissance de trajectoires humaines. Cependant, la partie concernant la conception de ces modèles est dirigée par des annotations réalisées *a priori* sur des trajectoires types d'apprentissage. Ce qui permet de constater qu'il n'existe pas réellement d'automatisme dans ce processus. De plus, les modèles ainsi établis forment une liste non exhaustive des activités que nous pouvons rencontrer dans la réalité.

Il est néanmoins envisageable d'utiliser cette méthode, permettant de modéliser, segmenter et classer des activités humaines, dans le but de générer des instances des activités apprises. Cependant, les activités étudiées représenteraient uniquement des événements simples tels que "s'arrêter devant la vitrine d'un magasin", "entrer et sortir du magasin",... Ce qui ne permet pas d'avoir un suivi précis sur des activités un peu plus complexes, exercées par les clients d'un magasin par exemple, et par conséquent, de générer et reproduire des comportements plus complets.

Plutôt que de se focaliser sur une segmentation en actions atomiques pour représenter des trajectoires, des techniques de classification automatique permettent des représentations uniques avec des trajectoires complètes.

2.4.4 Modèle de mélange pour la classification automatique

Comme nous pouvons le constater dans la section précédente, il est très courant d'utiliser l'information des trajectoires extraites à partir de séquences vidéo dans le but d'identifier certains comportements. Ceci s'explique par le fait que les trajectoires peuvent décrire différentes actions ou activités suivant la finalité de l'étude suggérée [Pusioli *et al.*, 2008]. Nous pouvons citer, par exemple, les trajectoires des mouvements de la main impliquées dans un processus de vote ; les trajectoires des voitures dans une course de Formule 1 [Hervieu *et al.*, 2009] ; les trajectoires des personnes suspectes d'actes de violence ou de comportements douteux [Wiliem *et al.*, 2009] ; les trajectoires des voitures dans une intersection [Morris et Trivedi, 2011] [Morris et Trivedi, 2008], la détection de trajectoires de circulation anormales ou irrégulières dans une zone urbaine [Piciarelli *et al.*, 2008] ou encore les trajectoires de consommateurs dans un magasin de vente (application présentée dans le premier chapitre),... Les domaines où l'analyse de trajectoire est pertinente sont donc variés et nombreux. Les méthodes et les modèles employés pour analyser les différentes trajectoires sont tout aussi divers.

Etant donné que, dans notre cas d'étude, nous nous intéressons à identifier les différents types de trajectoires de façon automatique sans annotation préalable (sans labellisation des familles des trajectoires par un expert sur des exemples observés), nous portons notre attention sur des méthodes impliquant des techniques d'apprentissage non supervisé. De ce fait, nous garantissons une extraction de comportements dépourvue de toute subjectivité. Nous présentons dans la suite différentes approches traitant de l'apprentissage non supervisé de comportements humains se basant sur les trajectoires.

En général, extraire des informations utiles à partir de données d'observation (trajectoires dans notre cas) non labellisées est réalisable au moyen de techniques de classification non supervisée ou automatique (clustering). En effet, elles sont utilisées pour regrouper les trajectoires non étiquetées en fonction de leur similarité. Ainsi, à partir d'un ensemble de trajectoires, ces techniques sont capables de créer des clusters (groupes) de telle sorte que celles appartenant au même groupe ont un potentiel

de grande similarité, et celles appartenant à deux groupes différents ont des similarités très faibles. Ainsi, les clusters, déterminés à partir des données d'observations réelles, peuvent être associés à des composantes du phénomène observé non aisément identifiables au préalable.

Nous nous intéressons, par conséquent, à une méthode particulière relative au problème de clustering de données trajectoires appliquant les modèles de mélanges de régressions [Gaffney et Smyth, 1999]. L'idée appliquée dans cette étude se rapproche, en quelque sorte, de celle que nous adopterons. L'analogie réside dans l'utilisation de l'algorithme EM pour l'estimation des paramètres en recherchant le maximum de vraisemblance.

Dans la perspective d'utiliser ce principe dans notre méthodologie, nous consacrons les deux sous-sections suivantes à introduire des notions fondamentales dont le maximum de vraisemblance et l'algorithme EM général, suivies de la méthode de classification des trajectoires développée par [Gaffney et Smyth, 1999] et basée sur un mélange de modèles de régression.

2.4.4.1 Le maximum de vraisemblance

L'algorithme EM utilise la notion de maximum de vraisemblance. Elle représente un critère permettant d'estimer les paramètres des distributions représentatives d'un échantillon de données. Dans le cadre de modèles de mélange gaussien, l'estimation du modèle de classification $\mathfrak{S}$, représentant l'ensemble des paramètres (π_i, θ_i) à estimer pour toutes les gaussiennes, recherche la partition optimale des données de sorte à former des classes homogènes $\mathcal{C}_M = \{C_m\}_{m=1,\dots,M}$ où M représente le nombre de modèles. En d'autres termes, cet estimateur cherche à maximiser la probabilité P d'appartenance des données $\chi_L = \{X_i\}_{i=1,\dots,L}$ aux classes où L représente le nombre d'observations. L'apparition du critère de la vraisemblance est ainsi obtenue par le produit des densités de probabilités des instances individuelles relativement à chacune des composantes $\theta_m = (\mu_m, \Sigma_m)$ du mélange où μ_m est la moyenne de la composante m tel que $1 \leq m \leq M$, Σ_m sa matrice de covariance et φ la densité de probabilité multidimensionnelle, définissant le modèle de mélange en tout point :

$$\ell(\chi_L, \mathfrak{S}) = \prod_{i=1}^L \varphi(X_i) = \prod_{i=1}^L \sum_{m=1}^M \pi_m \varphi(X_i | \theta_m) \quad (2.8)$$

Afin de faciliter la résolution du problème d'optimisation qui vise à rechercher le jeu de paramètres qui maximise cette vraisemblance, c'est la log-vraisemblance qui est exploitée et est exprimée par :

$$\log \ell(\chi_L, \mathfrak{S}) = \sum_{i=1}^L \log \sum_{m=1}^M \pi_m \varphi(X_i | \theta_m) \quad (2.9)$$

Annuler la dérivée de cette expression entraîne l'apparition des paramètres θ_m représentant l'estimation du modèle optimal $\mathfrak{S}_{opt}$ par la méthode de maximum de vraisemblance :

$$\mathfrak{S}_{opt} = \arg \max_{\mathfrak{S}} (\log \ell(\chi_L, \mathfrak{S})) \Rightarrow \frac{\partial \log \ell(\chi_L, \mathfrak{S}_{opt})}{\partial \mathfrak{S}_{opt}} = 0 \quad (2.10)$$

Cette manière d'estimer, de façon non supervisée, un modèle de classification est très prometteuse. Cependant, il est à noter que la dérivée de la log-vraisemblance n'est pas toujours dérivable et que celle-ci ne détient pas de solutions analytiques dans certains cas. En pratique, on utilise généralement cette technique dans un processus itératif estimant et maximisant ce critère. Ce principe fait l'objet de la section suivante.

2.4.4.2 L'algorithme EM

L'algorithme EM (Espérance-Maximisation) [McLachlan et Krishnan, 1996] est une technique de clustering qui cherche un modèle statistique représentant un modèle de mélange, qui décrive au mieux le jeu de données à segmenter, estimant ses paramètres grâce à une maximisation de la log-vraisemblance.

Il est basé sur un processus itératif qui cherche à maximiser cette vraisemblance jusqu'à convergence vers un optimum qui peut être local ou global.

Chaque itération effectue deux étapes :

- l'étape E (Espérance) qui calcule les probabilités *a posteriori* étant données les instances à classer. Cela revient à déterminer l'espérance conditionnelle de la vraisemblance à partir du modèle existant ;
- l'étape M (Maximisation) qui, comme son nom l'indique, maximise l'espérance de la vraisemblance et estime les paramètres du nouveau modèle.

L'algorithme 2 décrit un EM classique.

Algorithm 2 L'algorithme EM classique

Entrées : $\mathfrak{S}^0$: valeurs initiales des paramètres du modèle

χ_L : ensemble d'observations

ϵ : seuil de convergence

$k \leftarrow 0$

Répéter

Etape E (Espérance) : calcul de l'espérance conditionnelle de la vraisemblance complète :

$$Q(\mathfrak{S}|\mathfrak{S}^{(k)}) = E[\ell(X, \mathfrak{S})|\mathfrak{S}^{(k)}]$$

Etape M (Maximisation) : maximisation de l'espérance $Q(\mathfrak{S}|\mathfrak{S}^{(k)})$:

$$\mathfrak{S}^{(k+1)} = \arg \max_{\mathfrak{S}} Q(\mathfrak{S}|\mathfrak{S}^{(k)})$$

$k \leftarrow k + 1$;

Jusqu'à

$$\|Q(\mathfrak{S}|\mathfrak{S}^{(k)}) - Q(\mathfrak{S}|\mathfrak{S}^{(k+1)})\| \prec \epsilon;$$

Dans le cadre d'une étude de classification à l'aide d'un mélange gaussien, une variante de l'algorithme EM, nommée CEM (Conditional Expectation Maximization), est utilisée. Nous nous intéressons à cette version de EM car nous souhaitons l'appliquer ultérieurement dans notre méthodologie. Elle introduit un vecteur aléatoire $Z_L = \{z_i\}_{i=1,\dots,L}$ correspondant au vecteur d'appartenance d'un point x_i à la classe C_k .

Les réalisations $z_i = (z_{i,1}, \dots, z_{i,m})$ de Z_L sont indépendantes et identiquement distribuées selon la loi multinomiale à π résultats où $\pi = (\pi_1, \dots, \pi_M)^T$. Ces réalisations représentent les informations manquantes cachées sous forme de variables inconnues qu'il faut également estimer en même temps que les paramètres du modèle. Ceci permet de déterminer les classes d'appartenance des données.

Sachant que, dans le cas d'une partition dure, chaque donnée est affectée à une et une seule classe de telle sorte que les informations manquantes soient des variables binaires :

$$z_{i,m} = \begin{cases} 1 & \text{si } x_i \in \text{à la classe } C_m \\ 0 & \text{sinon.} \end{cases} \quad (2.11)$$

Soit $Y = (\chi_L, Z_L)$ les données complètes. On cherche, dans ce cas, à maximiser l'espérance conditionnelle de la vraisemblance complète :

$$\ell_c(\mathfrak{S}|Y) = P(X_L, Z_L|\mathfrak{S}) \quad (2.12)$$

Cela revient, comme mentionné dans la section précédente, à maximiser la log-vraisemblance :

$$\begin{aligned} \log \ell_c(\mathfrak{S}|Y) &= \log P(X_L, Z_L|\mathfrak{S}) \\ &= \log \left(\prod_{i=1}^L \varphi(x_i, z_i|\mathfrak{S}) \right) \\ &= \log \left(\prod_{i=1}^L \prod_{m=1}^M \pi_m \varphi(x_i, z_{i,m}|\mathfrak{S}_m) \right) \\ &= \sum_{i=1}^L \sum_{m=1}^M \log(\pi_m \varphi(x_i, z_{i,m}|\mathfrak{S}_m)) \end{aligned} \quad (2.13)$$

De ce fait, l'étape E de l'espérance de cette vraisemblance calcule la probabilité a posteriori et remplace les réalisations $z_{i,m}$ par l'estimateur de leur espérance. On a donc :

$$Z_{i,m} = E[Z_{i,m}|x_i, \mathfrak{S}_m^{(k)}] \quad (2.14)$$

D'où :

$$\begin{aligned} z_{i,m}^{(k)} &= P(Z_{i,m} = m|x_i, \mathfrak{S}_m^{(k)}) \\ &= \frac{P(Z_{i,m} = m, x_i|\mathfrak{S}_m^{(k)})}{P(x_i|\mathfrak{S}_m^{(k)})} \\ &= \frac{\pi_m^{(k)} \varphi(x_i|\mathfrak{S}_m^{(k)})}{\sum_{m=1}^M \pi_m^{(k)} \varphi(x_i, \mathfrak{S}_m^{(k)})} \end{aligned} \quad (2.15)$$

A partir de cette équation, il est possible de calculer les probabilités a posteriori du modèle. De plus, dans le cas de mélange de composantes gaussiennes, le modèle théorique étant connu, la phase de maximisation peut être décrite en estimant les paramètres π_m , μ_m et Σ_m de la classe m . Pour la version classificatoire, les étapes de l'espérance et de la maximisation dans le cas de modèles gaussiens sont décrites dans l'algorithme 3. L'estimation des paramètres du modèle de mélange gaussien, ainsi obtenue, représente la partition optimale des données par rapport à l'homogénéité des classes de la partition.

Malgré l'efficacité de cet algorithme en termes de résultats, il y persiste quelques défauts constatés. En effet, l'algorithme EM peut converger vers un optimum local dépendant de l'initialisation du modèle. Garantir la convergence vers un minimum global pourrait être un processus très lent. De plus l'algorithme dans sa version classificatoire peut être également sensible aux données aberrantes et le nombre de classes doit être fixé au préalable.

Algorithm 3 L'algorithme EM pour les modèles de mélanges gaussiens**Entrées :** X : ensemble d'observations M : le nombre de composantes du mélange $\mathfrak{S}^{(0)} = [\pi_1^0, \dots, \pi_M^0, \mu_1^{(0)}, \Sigma_1^{(0)}, \dots, \mu_1^{(M)}, \Sigma_1^{(M)}]$: une valeur initiale des paramètres du modèle ϵ : seuil de convergence $k \leftarrow 0$ **Répéter****Etape E** (Espérance) : calcul des probabilités *a posteriori* :

$$z_{i,m}^{(k)} = \frac{\pi_m^{(k)} \varphi(x_i | \mathfrak{S}_m^{(k)})}{\sum_{m=1}^M \pi_m^{(k)} \varphi(x_i, \mathfrak{S}_m^{(k)})}$$

Etape C (Classification) : affectation des données aux classes :

$$\begin{cases} z_{i,m}^{(k)} = 1 & \text{si } z_{i,m}^{(k)} = \arg \max z_{i,m}^{(k)} \quad \text{i.e. } (X_i \in C_m) \\ z_{i,m}^{(k)} = 0 & \text{sinon} \quad \text{i.e. } (X_i \notin C_m) \end{cases}$$

Etape M (Maximisation) : estimation des paramètres π_m , μ_m et Σ_m de la classe m :

$$\pi_m^{(k+1)} = \frac{\sum_{i=1}^L z_{i,m}^{(k)}}{L}$$

$$\mu_m^{(k+1)} = \frac{\sum_{i=1}^L z_{i,m}^{(k)} x_i}{\sum_{i=1}^L z_{i,m}^{(k)}}$$

$$\Sigma_m^{(k+1)} = \frac{\sum_{i=1}^L z_{i,m}^{(k)} (x_i - \mu_m^{(k+1)})(x_i - \mu_m^{(k+1)})^T}{\sum_{i=1}^L z_{i,m}^{(k)}}$$

 $k \leftarrow k + 1$;**Jusqu'à**

$$\|Q(\mathfrak{S} | \mathfrak{S}^{(k)}) - Q(\mathfrak{S} | \mathfrak{S}^{(k+1)})\| < \epsilon;$$

Ayant présenté l'algorithme EM, nous allons étudier une technique particulière réalisée pour le clustering des trajectoires avec les mélanges de modèles de régression.

2.4.4.3 EM pour les mélanges de modèles de régression

Les données d'entrée considérées ici sont des trajectoires extraites de séquences vidéo. L'idée est d'obtenir à partir de ces trajectoires, des clusters dont chacun est modélisé par une fonction lisse d'une variable indépendante x , représentant le temps, avec un bruit additif utilisant le principe de régression. Ainsi, chaque fonction lisse est estimée à partir des mélanges de modèles de régression dont les paramètres sont appris suivant l'algorithme EM [Gaffney et Smyth, 1999].

Supposons qu'on dispose d'un ensemble de données y et x connu (pour une question de simplicité x et y sont unidimensionnels). La relation de régression standard entre x et y est définie comme suit :

$$y = g_k(x) + e_k \quad (2.16)$$

où e_k est un bruit blanc gaussien de moyenne zéro et d'écart type σ_k et g_k est une fonction déterministe de x de la composante k .

A l'instar du modèle de mélange gaussien, chaque composante est une fonction de densité conditionnelle $\varphi_k(y|x, \theta_k)$ telle que y est une conséquence du $k^{\text{ème}}$ modèle, d'un bruit blanc de variance σ_k , et θ_k représente les paramètres du modèle $g_k(x)$ et σ_k .

Vu que l'appartenance des trajectoires aux groupes est inconnue, le recours à l'emploi de l'algorithme EM est nécessaire. En effet, grâce à son processus itératif, les données cachées correspondant à l'appartenance de chaque trajectoire à un groupe sont réestimées jusqu'à obtenir une convergence telle que la $\log\ell(\theta|S)$ ne diminue quasiment pas, avec S l'ensemble des données composé de n_j mesures pour chacune des M trajectoires tel que $1 \leq j \leq M$. L'équation de la log-vraisemblance des paramètres θ des K composantes, dans ce cas de figure, ayant un ensemble de données trajectoires S et Z la matrice des vecteurs indicateurs $z_j = (z_{j1}, \dots, z_{jK})$ (correspond au Z de la section 2.4.4.2), est définie comme suit :

$$\log\ell(\theta|S, Z) = \sum_j \sum_k z_{jk} \log \pi_k + \sum_j \sum_k \sum_i^{n_j} z_{jk} \log \varphi_k(y_j(i)|x_j(i), \theta_k) \quad (2.17)$$

Où $y_j(i)$ représente la $i^{\text{ème}}$ mesure du $j^{\text{ème}}$ individu générée par $x_j(i)$ et $z_{jk} = 1$ si la $j^{\text{ème}}$ trajectoire est considérée comme générée par la $k^{\text{ème}}$ composante.

Posons $h_{jk} = E[z_{jk}]$ et $Y_j = X_j \beta_k + e_k$ l'équation de régression des données S , avec :

$$Y_j = \begin{bmatrix} 1 & y_j(1) & \cdots & y_j(n_j) \end{bmatrix}'$$

$$\text{et } X_j = \begin{bmatrix} 1 & x_j(1) & x_j(1)^2 & \cdots & x_j(1)^p \\ 1 & x_j(2) & x_j(2)^2 & \cdots & x_j(2)^p \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_j(n_j) & x_j(n_j)^2 & \cdots & x_j(n_j)^p \end{bmatrix},$$

avec p l'ordre du modèle de régression.

La première partie de l'algorithme EM estimera l'équation suivante :

$$h_{jk} = \pi_k \prod_i^{n_j} \varphi_k(y_j(i)|x_j(i), \theta^{t-1}) \quad (2.18)$$

La deuxième partie de l'algorithme maximisera les équations suivantes :

$$\hat{\beta}_k = (X' H_k X)^{-1} X' H_k Y \quad (2.19)$$

$$\hat{\sigma}_k^2 = \frac{(Y - X \beta_k)' H_k (Y - X \beta_k)}{\sum_j^M h_{jk}} \quad (2.20)$$

$$\hat{\pi}_k = \frac{1}{M} \sum_j^M h_{jk} \quad (2.21)$$

Tel que les paramètres $\theta_k = \{\pi_k, \beta_k, \sigma_k^2\}$ représentent respectivement, les poids du mélange, les coefficients de régression et les variances. Avec $H_k = \text{diag}([h_{1k}^* \ h_{2k}^* \ \dots \ h_{Mk}^*])$ tels que $h_{jk}^* = [h_{jk}^{(1)} \ h_{jk}^{(2)} \ \dots \ h_{jk}^{(n_j)}]$. Si on considère $N = \sum_j^M n_j$ alors H_k est une matrice diagonale $N \times N$ dont les éléments de sa diagonale représentent les poids à appliquer à Y et X durant la régression, sachant que $Y = [Y_1' \ Y_2' \ \dots \ Y_M']'$ représentant toutes les mesures de $y_j(i)$ de chaque trajectoire (Y est une matrice $N \times 1$) et $X = [X_1' \ X_2' \ \dots \ X_M']'$ dont la seconde colonne correspond aux points temps quand les valeurs de Y correspondantes ont été mesurées (X est une matrice $N \times p + 1$).

La figure 2.8 montre des résultats, empruntés aux travaux de [Gaffney et Smyth, 1999], de classification de trajectoires, synthétiques, suite à l'emploi de la méthode de mélanges de modèles de régression. Cette technique a également été testée sur un flux de données vidéo représentant un mouvement de bras. Il s'agit notamment d'un mouvement vertical ou d'un mouvement horizontal pouvant identifier certaines activités tel que le vote par exemple.

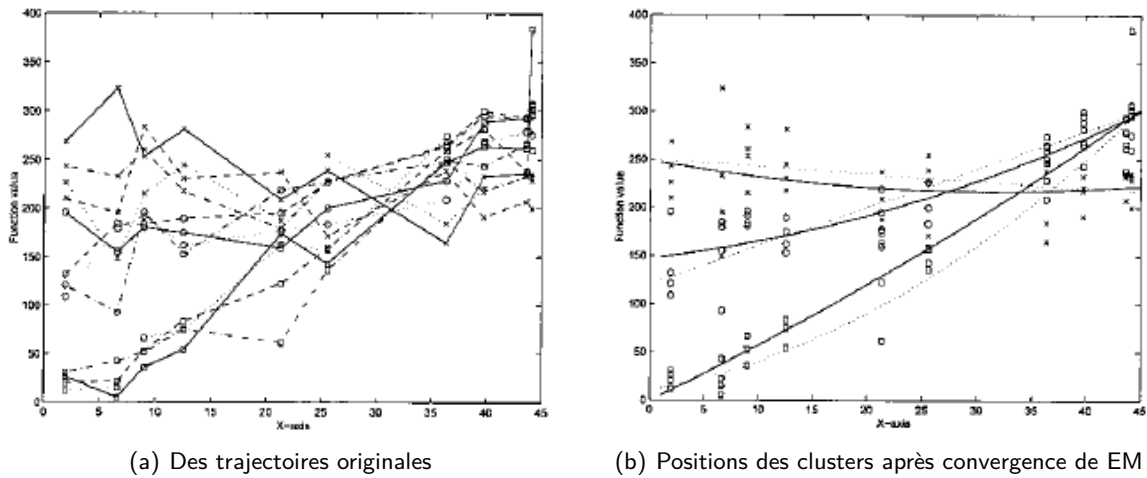


FIGURE 2.8 – Exemple de résultats de la méthode de mélanges de modèles de régression, extrait de [Gaffney et Smyth, 1999]

Dans [Gaffney et Smyth, 1999], des tests comparatifs entre cette méthode et la méthode du mélange de modèles gaussiens ont été réalisés afin d'évaluer la performance de l'une par rapport à l'autre. Il a été démontré au bout de quelques exemples, utilisant des données à une dimension, que la méthode des mélanges de modèles de régression est légèrement plus efficace. Néanmoins, la performance des résultats obtenus varie d'un exemple de test à un autre, en agissant sur le nombre de données d'entrée ou sur les paramètres des clusters (l'écart type du bruit par exemple). De plus, une comparaison à N dimensions n'a pas été démontrée.

2.5 Discussion et conclusion

Dans ce chapitre, des méthodes d'extraction de connaissances, à partir de données provenant de l'observation (séquences vidéo), ont été abordées. L'objectif est d'extraire des informations utiles pour la conception d'agents, le plus objectivement possible, en limitant le recours à des experts et à des enquêtes empiriques. Afin de simuler un monde réel complexe à partir de son observation, nous nous sommes focalisés sur les trajectoires des agents physiques. Celles-ci permettent d'apporter un certain nombre de connaissances pertinentes sur des comportements de déplacement ou des événements se déroulant dans l'environnement à reproduire.

Parmi les méthodologies existantes portant sur l'analyse de données vidéo pour l'extraction de comportements de déplacement, nous avons exposé deux catégories différentes.

La première catégorie d'extraction de comportements de déplacement porte sur la conception de structures à états-transitions. Ces dernières permettent de décrire des activités types, en l'occurrence des mouvements de déplacement spécifiques, à partir de certaines actions ou fonctions détectées depuis des séquences vidéo. Les structures modélisées sont construites suivant différents mécanismes dont certains se basent sur les modèles dynamiques commutés, d'autres sur les réseaux bayésiens dynamiques récurrents, ainsi que sur les modèles de Markov cachés. Ces méthodes sont principalement utilisées pour la reconnaissance d'activités, ce qui requiert généralement des annotations manuelles préalables et donc la présence d'un superviseur humain.

Outre le fait que l'usage des MMC est, en majorité, destiné à des applications de reconnaissance à différents niveaux et sous différentes formes, il correspond en général à un et un seul comportement spécifique. Il serait cependant intéressant d'explorer des situations où un MMC est capable d'apprendre et générer différents comportements en même temps. De cette manière, il serait possible d'apprendre, par un MMC, des comportements de déplacement initialement sous forme de trajectoires. A partir du MMC obtenu, il serait envisageable de générer des comportements de déplacements dont les trajectoires seraient assez proches de celles réelles. Ainsi, le MMC conçu aurait pour vocation non pas de traiter des problèmes de reconnaissance, mais de reproduire (générer) des séquences de déplacement. Dans cette optique, nous reviendrons dans les chapitres suivantes sur les MMC en considérant cette fois un cadre non-supervisé.

La seconde catégorie d'extraction de comportements de déplacement concerne les techniques de classification automatique. Certaines de ces techniques, dont l'algorithme EM, permettent d'éviter le recours à des annotations et d'identifier de façon non supervisée différents types de trajectoires. Nous avons présenté, de ce fait, la structure de cet algorithme ainsi que la méthode de mélanges de modèles de régression, traitant de la classification de trajectoires, dont les paramètres sont appris selon l'algorithme EM.

Le chapitre suivant présente notre contribution dans laquelle nous explorons, dans un contexte non supervisé, l'utilisation d'une part de l'algorithme EM et d'autre part des MMC pour la construction automatique de comportements d'agents à partir de trajectoires extraites de séquences vidéo.

Chapitre 3

Méthodologies et contribution

Sommaire

3.1	Introduction	49
3.2	Méthodologie de conception de simulation multi-agents	51
3.2.1	Prérequis	54
3.2.2	Comportements individuels des agents	55
3.2.3	Données collectives observées pour la vérification du modèle	61
3.3	Extraction de comportements à partir d'observations	63
3.3.1	Application de l'algorithme EM	63
3.3.2	Application des MMC	73
3.4	Discussion et conclusion	79

3.1 Introduction

Nous avons exposé, dans les chapitres précédents, les difficultés rencontrées lors de la conception d'une simulation multi-agents reproduisant des comportements observés. Nous nous sommes focalisés, en particulier, sur un cas d'application bien défini qui représente la simulation de déplacements de consommateurs dans un espace de vente. Les difficultés se regroupent en deux problématiques :

- Quelle méthodologie de conception de simulation multi-agents est la plus appropriée pour reproduire des comportements de déplacement dans un environnement donné ?
- Comment parvenir à extraire, à partir de l'observation du monde réel, des informations et des paramètres permettant de concevoir une simulation reproduisant fidèlement la réalité ?

Dans ce chapitre, nous développons une proposition qui répond aux deux points cités ci-dessus. En effet, elle porte sur la conception de simulation d'agents situés, dans un environnement connu. Cette proposition repose sur l'hypothèse que le concepteur de la simulation dispose de connaissances préalables sur l'environnement. Par exemple, s'il souhaite simuler un supermarché, il lui sera nécessaire d'avoir à sa disposition le plan du local comprenant la répartition des linéaires ainsi que les produits correspondants.

Dans notre proposition, les agents reproduisent des comportements réels, extraits à partir de traitements de séquences vidéo. En première approche, nous nous sommes restreints à des comportements uniquement basés sur des déplacements d'individus. Les trajectoires ainsi extraites permettent de révéler des informations pertinentes par rapport à l'attractivité des différentes régions de l'environnement. Dans notre travail, cette attractivité peut être visualisée et évaluée facilement grâce aux cartes des points chauds et des points froids résultantes de la simulation multi-agents.

L'intérêt de simuler de tels déplacements, construits sur la base de trajectoires observées, réside dans le fait de pouvoir agir, en simulation, sur l'environnement en modifiant ses propriétés et de visualiser le résultat engendré par cette modification. Prenons, par exemple, un cas de figure très simple où on cherche à simuler des trajectoires accomplies par des consommateurs dans un petit espace de vente durant une période de soldes. Supposons que ces clients s'intéressent uniquement aux produits soldés placés au fond du magasin. Les clients vont se déplacer jusqu'à l'emplacement des produits désirés puis revenir vers la caisse se trouvant à l'entrée du magasin. Connaissant l'environnement et le but des clients (les articles soldés par ex) grâce à leurs trajectoires, il est donc possible de reproduire ces comportements par des agents ayant le même but dans une simulation du magasin. Le résultat de la simulation serait une carte des points chauds et des points froids démontrant une concentration (une densité importante de points chauds) au fond du magasin et de légères concentrations au niveau des flux de passages entre l'entrée et les produits soldés. Or si on modifie, dans la simulation, l'emplacement des articles convoités en les plaçant à l'entrée du magasin (modification des propriétés de l'environnement), les agents suivront des trajectoires différentes par rapport aux initiales. La carte des points chauds et des points froids ainsi obtenue (résultat engendré par la modification), aura des propriétés différentes, montrant l'impact possible du nouvel aménagement sur les flux et la répartition géographique des consommateurs.

Bien sûr, la réalité est beaucoup plus complexe. Il existe une multitude de comportements différents, notamment des clients intéressés par la nouvelle collection, des clients cherchant un article en particulier, des clients adeptes du lèche-vitrine, etc. Reproduire individuellement toutes les trajectoires ainsi observées manquerait de généralisation. Or, elles présentent de multiples similarités qui peuvent être exploitées pour regrouper les trajectoires similaires ou proches. Les groupes ainsi obtenus peuvent être considérés comme différents profils d'individus observés. La simulation ainsi obtenue à partir de ces profils représente une simplification et une abstraction du monde réel tout en étant suffisamment réaliste pour être exploitée (ie assez réaliste pour permettre d'agir aisément sur les composantes du système simulé et prévoir les résultats du monde réel).

La première section de ce chapitre présente les différentes étapes de notre méthodologie de conception de simulation multi-agents. Celle-ci est illustrée au travers d'un exemple facilitant la compréhension de notre démarche. Nous utilisons le hall d'entrée d'un établissement universitaire comme cadre applicatif afin d'étudier la faisabilité de notre approche sur un exemple délibérément simple et qui pourrait être, par la suite, facilement transposée à un espace de vente. La seconde section de ce chapitre se focalise sur les méthodes d'apprentissage utilisées dans notre proposition.

3.2 Méthodologie de conception de simulation multi-agents

Un processus de simulation multi-agents consiste à reproduire le comportement dynamique d'un phénomène réel en utilisant des modèles d'agents et différentes séquences d'entrée. Le modèle d'agent est une construction simplifiée de la réalité qui peut être utilisée pour reproduire, selon un certain niveau de détail, un comportement d'un élément du système à simuler. Comme indiqué dans le chapitre 1, le comportement individuel des agents, ainsi modélisés, et leurs interactions dans un environnement ciblé, permettent de reproduire la dynamique globale du système.

La mise en œuvre d'une modélisation "agent" n'étant pas triviale, de nombreux efforts ont été fournis pour proposer des méthodologies facilitant la conception des systèmes multi-agents. Nous trouvons, dans la dernière décennie, de nombreuses méthodes de modélisation agent ; pour n'en citer que quelques-unes : Gaia [Wooldridge *et al.*, 2000], O-MAsE [García-Ojeda *et al.*, 2007], INGENIAS [Pavón et Gómez-Sanz, 2003], ADELFE [Bernon *et al.*, 2003]. Tous ces travaux donnent un cadre général pour concevoir un système multi-agents, y compris : des modèles prédéfinis d'agents abstraits, des schémas d'organisation, des protocoles de communication, des actes de langage, etc. Les directives associées à ces méthodologies permettent de raffiner itérativement ces modèles abstraits en des modèles concrets, jusqu'à l'implémentation des agents. Bien que ces méthodologies puissent être appliquées à la conception des simulations multi-agents, elles ne sont pas bien adaptées à la construction de modèles fidèles à une réalité observée, principalement parce qu'elles ont un point de vue global et fonctionnel du système multi-agents. Des approches spécifiques pour la conception des simulations multi-agents ont été introduites offrant une abstraction plus appropriée et des architectures pour exprimer les comportements et les interactions des agents. Par exemple, dans IODA [Kubera *et al.*, 2011] présenté dans le premier chapitre, les auteurs proposent de construire une simulation reposant sur les différentes interactions entre les agents. Il est clair qu'une telle approche facilite la modélisation et la mise en œuvre des agents dans des simulations multi-agents, mais ne dispense pas le concepteur d'une étape ardue : l'observation et l'analyse des phénomènes à simuler dont les comportements.

Cette étape est habituellement traitée par des experts du domaine (comme les sociologues ou psychologues) afin d'obtenir des modèles abstraits dont les implémentations seront réalisées par les développeurs. En règle générale, cette étape est très coûteuse en temps, car elle nécessite des études pouvant reposer par exemple sur le traitement de données statistiques, l'analyse de séquences vidéo, le dépouillement de questionnaires, l'interview d'experts, ... En plus d'être fastidieuse, cette tâche est très subjective et les modèles résultants peuvent varier d'un expert à l'autre. Par ailleurs, les méthodes citées ci-dessus n'évitent pas les problèmes de compréhension entre experts et développeurs. Une façon de réduire tous ces désagréments est d'automatiser autant que possible l'étape d'observation et d'analyse. Pour ce faire, les outils d'apprentissage non supervisé semblent être une bonne approche pour fournir des connaissances facilitant la conception des simulations multi-agents.

Comme nous l'avons vu dans les chapitres précédents, l'intégration de techniques de data mining en général aux simulations multi-agents est essentiellement spécifique aux phases de raffinement (amélioration) et de validation des modélisations existantes. Nous avons souhaité aller plus loin et utiliser des techniques de data mining sur les données d'observation pour faciliter la modélisation du

comportement des agents. Deux méthodes permettant de réaliser cette modélisation ont été étudiées. Il s'agit de :

- la segmentation (ou classification non supervisée) également plus connue sous le terme anglophone "Clustering",
- les modèles de Markov cachés (MMC).

Comme nous avons pu le remarquer au deuxième chapitre, ces deux méthodes traitent les données de façon différente. Néanmoins, chacune d'elles présente un certain nombre de qualités que nous souhaitons évaluer. Dans ce qui suit, nous soulignons, pour chaque technique citée respectivement, les propriétés pertinentes pour notre étude.

Utilité des techniques de clustering

Sous l'hypothèse de disposer de données d'observation du ou des phénomènes à simuler, les techniques de segmentation peuvent en extraire des informations utiles pour :

- identifier les différentes entités composant le système à simuler (fonction de leur importance) ;
- concevoir les différents types de comportement d'agents ;
- répartir ces agents au sein de la simulation.

Parmi les techniques utilisées pour l'exploration de données, les techniques de segmentation permettent de regrouper les données non étiquetées en fonction de leurs similarités. Ainsi, à partir d'un ensemble de données, ces techniques sont capables de créer des clusters (groupes) de telle sorte que les données appartenant au même groupe aient une grande similarité, et les données appartenant à deux groupes différents soient dissemblables.

Ainsi, les clusters, déterminés à partir des données d'observations réelles, peuvent être associés à différents agents à modéliser. En d'autres termes, plutôt que de se baser sur sa propre expérience pour identifier les entités essentielles à modéliser, le concepteur peut s'appuyer sur les clusters ainsi obtenus.

A chaque cluster, sont attachées diverses informations ou mesures telles que l'indice, le centre (également appelé prototype du groupe), la cardinalité, la variance, la densité,... Certaines de ces valeurs peuvent être interprétées dans le contexte de la modélisation des agents [Saffar *et al.*, 2011] :

- l'indice d'un cluster permet d'identifier un type d'agent particulier ;
- le prototype du groupe fournit une information moyenne représentative des données présentes au sein du cluster. Cette information peut être ensuite utilisée pour formaliser le comportement du type d'agent associé à ce cluster ;
- la cardinalité (les distributions des données au sein de chaque cluster) donne la répartition des agents par type au sein de la simulation.

Utilité des MMC

Admettons que nous disposons de quantités de données d'observation suffisantes pour apprendre un MMC. Celui-ci peut révéler un graphe d'états cachés englobant l'ensemble des comportements observés. Selon la façon dont le graphe est parcouru, des comportements différents pour les agents

peuvent être générés. Ainsi, la diversité des agents, conçus à partir de ce graphe, est vaste. Contrairement à la méthode de clustering, il n'existe pas de type spécifique d'agent.

Les deux techniques d'apprentissage que sont le clustering et le MMC semblent intéressantes à expérimenter. Nous les avons donc appliquées et comparées dans le cadre de notre proposition. Toutefois, elles ne représentent qu'une brique de notre méthodologie de conception de simulation multi-agents. Ainsi, notre méthodologie est composée de plusieurs modules illustrés dans la figure 3.1. Nous y distinguons deux phases principales :

- la construction des modèles d'agent.
- la validation.

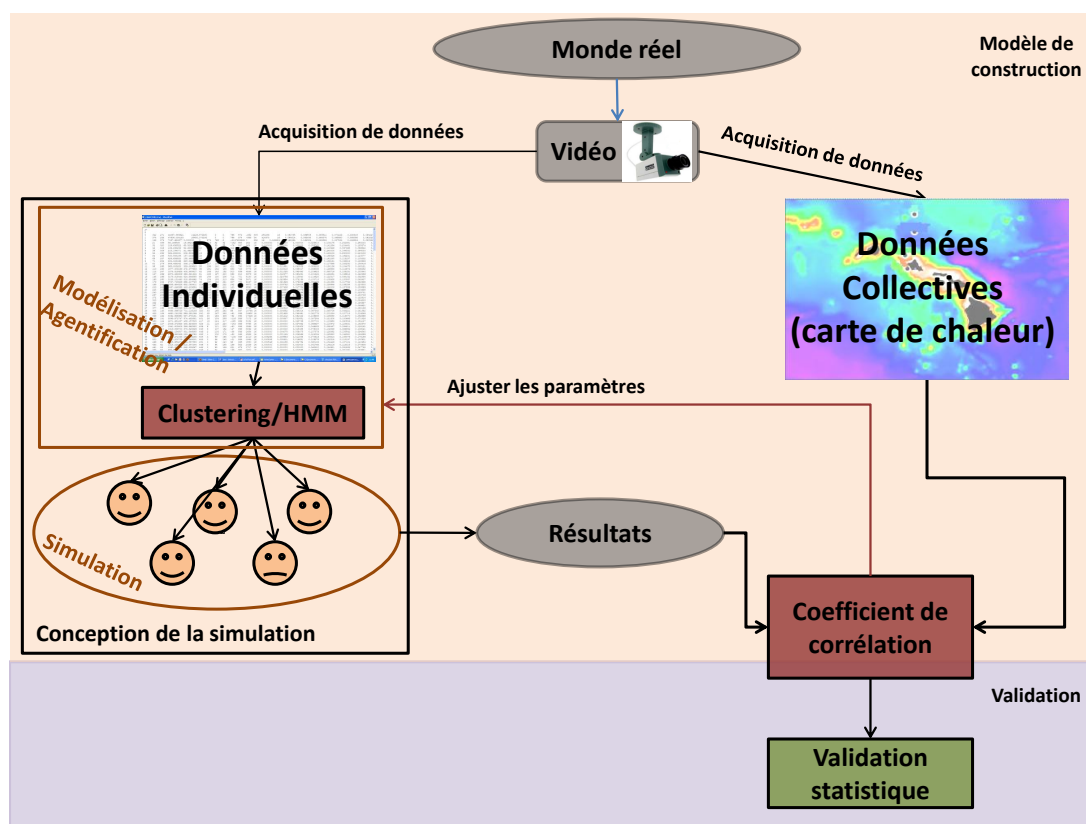


FIGURE 3.1 – Méthodologie de construction de simulation multi-agents

La première étape repose principalement sur l'utilisation de données individuelles sur lesquelles peuvent être appliquées des techniques d'apprentissage (clustering ou MMC) comme discuté ci-dessus. Ces données contribuent ainsi à définir la représentation sous forme d'agents de différentes typologies d'individus observés dans un environnement spécifique.

La deuxième étape nécessite quant à elle des données collectives. Ces données peuvent être par exemple une carte de chaleur mettant en valeur les régions les plus fréquentées et les plus désertes, ou encore des mesures caractérisant les flux (débit),... En les comparant avec les résultats de simulation, ces données collectives sont classiquement utilisées dans l'étape de validation du modèle de

simulation. Suivant les résultats de cette comparaison, il est possible d'agir sur les modèles et les traitements en amont de la simulation afin de les ajuster.

Selon le domaine d'application, l'étape de validation peut également impliquer un expert du domaine. A partir de son expérience et de ses connaissances, il valide ou non le réalisme et la pertinence de la simulation obtenue. Cependant, il intervient uniquement en tant que juge pour donner son approbation à l'approche appliquée à son domaine spécifique, et n'intervient pas directement dans la conception de la simulation.

Notre approche est relativement générique et peut s'appliquer à la simulation de tous comportements humains spatialisés dans un environnement connu, pour peu que ces derniers soient modélisables au travers des différentes observations (trajectographies,...). Les sections suivantes donnent en détail la démarche générale de l'approche en l'illustrant avec l'exemple applicatif choisi de façon à illustrer simplement notre méthodologie.

3.2.1 Prérequis

L'approche décrite ci-dessus a été mise en œuvre expérimentalement sur un cas d'étude impliquant la simulation d'activités humaines dans un hall d'un bâtiment universitaire. Celui-ci se permet l'accès à trois entrées, deux salles de classe, plusieurs bureaux, un escalier, un espace détente et une machine à café.

L'objectif est de reproduire, par simulation, les activités se déroulant au sein de cet espace : la pause pour les étudiants et le personnel, les passages dans le hall, les entrées et les sorties des classes, etc. Une caméra placée au-dessus du hall d'entrée nous permet d'enregistrer des séquences vidéo. A travers la simulation de cet environnement observable, on souhaite reproduire la densité d'occupation du hall. La figure 3.2 représente l'environnement étudié tel qu'il est perçu par la caméra.



FIGURE 3.2 – Environnement étudié

La première étape de notre approche est de traiter les séquences vidéo en extrayant les trajectoires individuelles qui serviront d'entrée pour l'algorithme de clustering ainsi que pour le modèle MMC. À partir du flux vidéo, l'algorithme de traitement extrait les formes en mouvement de l'image. Cette première opération est effectuée sur la base d'une soustraction de fond. Pour chaque blob (un ensemble de pixels représentant un corps humain) extrait et identifié, un vecteur représentant la trajectoire est alors construit, à partir du déplacement de la personne. Chaque trajectoire est caractérisée par un index unique et se compose d'une liste de points estampillés par un numéro de trame (fournissant une indication temporelle) et différents paramètres décrivant le blob (taille, forme, vecteur d'orientation, etc).

Servant de base à notre étude, nous avons enregistré des séquences vidéo du hall d'entrée du département durant une journée entière et en avons extrait 748 trajectoires de taille variable. Les plus courtes, qui correspondent à des déplacements non pertinents, ont été éliminées. Effectivement, les trajectoires très courtes correspondent dans la plupart des cas à un bruit dû aux effets de la lumière ou à une limitation du traitement vidéo qui, dans certaines circonstances, ne permet plus de suivre les blobs. Nous sélectionnons, de ce fait, les trajectoires qui sont composées d'au moins 30 points. Pour être en mesure d'effectuer une modélisation de l'ensemble de ces trajectoires dans un espace fixe, nous avons été amenés à ré-échantillonner toutes les trajectoires en considérant la taille moyenne des trajectoires égale à 50 points.

Enfin, nous avons obtenu 682 trajectoires représentées dans une matrice où chaque ligne contient une observation (un chemin) composée de 100 attributs : 50 éléments de l'axe des abscisses alternés avec 50 éléments de l'axe des ordonnées. Cet ensemble de trajectoires constitue notre base de données de référence à partir de laquelle sera présentée notre méthodologie.

Les trajectoires prétraitées étant disponibles sous forme d'un ensemble d'instances, il est envisageable d'y appliquer les traitements d'apprentissage non supervisé, dans le but de produire les futurs comportements de déplacement individuels des agents. Nous détaillons cette approche dans ce qui suit.

3.2.2 Comportements individuels des agents

Dans un environnement quelconque, nous nous apercevons que chaque individu possède son propre comportement. Dans un supermarché, par exemple, nous pouvons rencontrer un consommateur pressé, un autre qui se promène un peu partout entre les linéaires, un autre qui cherche les promotions ou un produit en particulier,... De cette diversité riche et complexe, il est souhaitable de trouver des modèles suffisamment simples et réalistes permettant de représenter des groupes de personnes ayant des comportements similaires.

Revenons à notre environnement d'application. La figure 3.3 montre le déplacement d'un individu suivant une trajectoire d'est en ouest. Compte tenu de l'aménagement du hall, ce déplacement est relativement récurrent puisqu'il correspond à une traversée du hall pour passer de l'aile est à l'aile ouest du bâtiment. Il existe donc un grand nombre de personnes ayant le même but que l'individu en question et qui, par conséquent, emprunte le même chemin à quelques disparités près. Il en est de même pour d'autres trajectoires telles que le chemin opposé par exemple, ou la trajectoire allant de l'entrée principale et à la machine à café, etc.



FIGURE 3.3 – Exemple d'une trajectoire suivie par un individu sur les images de la vidéo

Supposons que nous ayons à disposition toutes les trajectoires possibles dans notre environnement d'étude. Comment procède-t-on pour intégrer ces comportements de déplacement dans les agents au sein d'une simulation reproduisant le plus exactement possible les mouvements au sein de cet environnement? L'idée est de traduire ces trajectoires en motivations (objectifs). L'individu de la figure 3.3, par exemple, a pour but d'accéder directement au couloir de l'aile est. Par conséquent, il possède une succession d'objectifs relatifs aux emplacements auxquels il doit se rendre jusqu'à la destination finale. Cette abstraction des trajectoires en motivations nécessite un partitionnement de l'espace en différentes zones. Cette étape fait l'objet de la section suivante.

3.2.2.1 Discrétisation et annotation de l'environnement observé et simulé

Ayant à disposition le plan du hall et connaissant la fréquentation des différentes zones du hall, nous sommes capables d'identifier des régions d'intérêt (RI) (machine à café, couloirs, entrées de bureau, entrées de salle de cours, escalier, ...). Nous associons par la suite, à chaque région, un label qui la caractérise. La figure 3.4 représente le fond de la séquence vidéo annoté d'une partie de ces labels.

D'autre part, nous reproduisons au niveau de la simulation le même environnement vu à l'aplomb. Pour simplifier l'implémentation, l'environnement a été discrétisé sous forme d'une grille avec une granularité relativement fine par rapport à l'espace moyen occupé par un individu (voir figure 3.5 à droite). Nous reportons, par la suite, les labels des RI de l'environnement réel aux cellules corres-

pondantes de la simulation. L'espace est, par conséquent, partitionné en 70 zones dont chacune est étiquetée par un label correspondant à une RI. Elles sont représentées dans la figure 3.5 à gauche. Certains de ces labels ont une signification explicite telle que "Coffee1" pour désigner une première zone où se situe la machine à café. Ils représentent donc les régions d'intérêt principales dont la liste exhaustive est donnée ci-après : Exit, Stairs, Board1, Board2, Board31, Board32, Board33, Board34, Board41, Board42, Chair, Chair1, Chair2, Chair3, Chair4, Chair5, Chair6, Chair7, Table1, Table2, Corridor1, Corridor21, Corridor22, Corridor23, Corridor24, Office1, Office2, Trash, Coffee1, Coffee2, Printer, ClassEntry11, ClassEntry12, ClassEntry21, ClassEntry22, ClassEntry23. D'autres, principalement situées au milieu de l'environnement, ne représentent pas de véritables régions d'intérêt mais sont tout de même utiles à la modélisation d'objectifs intermédiaires. Ces régions ont des labels étiquetés de "C0" à "C33".

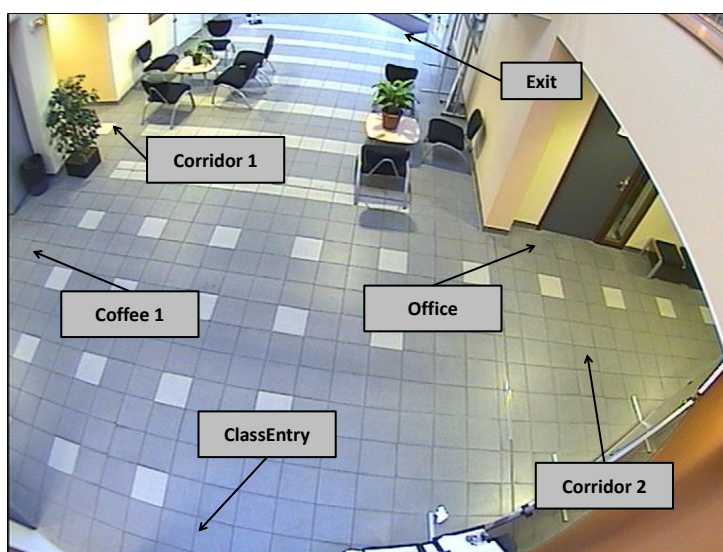


FIGURE 3.4 – L'environnement réel

Il est toutefois utile de constater que le découpage de l'environnement observé n'est pas uniforme et assimilable à un quadrillage régulier. Ce découpage résulte en effet d'une transformation géométrique permettant de passer du champ visuel de la caméra à une vue à l'aplomb de l'environnement et corrigeant les effets de perspective et de distorsions.

3.2.2.2 Motivations des agents à partir de l'environnement observé

Le principe qui sera suivi consiste à représenter les trajectoires des individus, circulant dans l'environnement discrétisé, sur la base des étiquettes des RI. En effet, considérons, une fois de plus, le chemin réalisé par l'individu de la figure 3.3. Selon le partitionnement de l'environnement, sa trajectoire est caractérisée par un ensemble de segments parcourant les régions d'intérêt (voir figure 3.6).

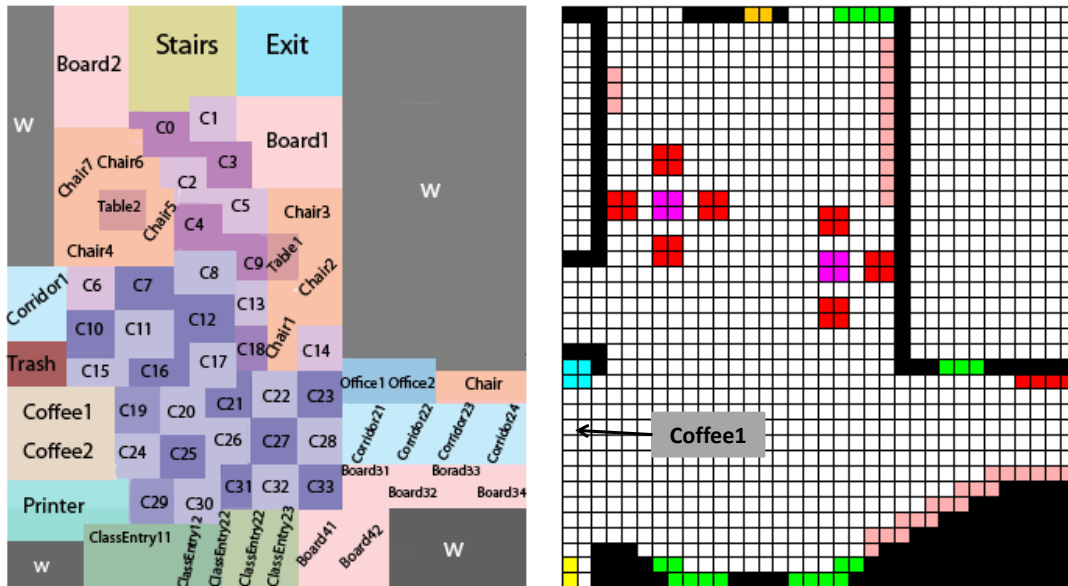


FIGURE 3.5 – Grille de l'environnement de la simulation

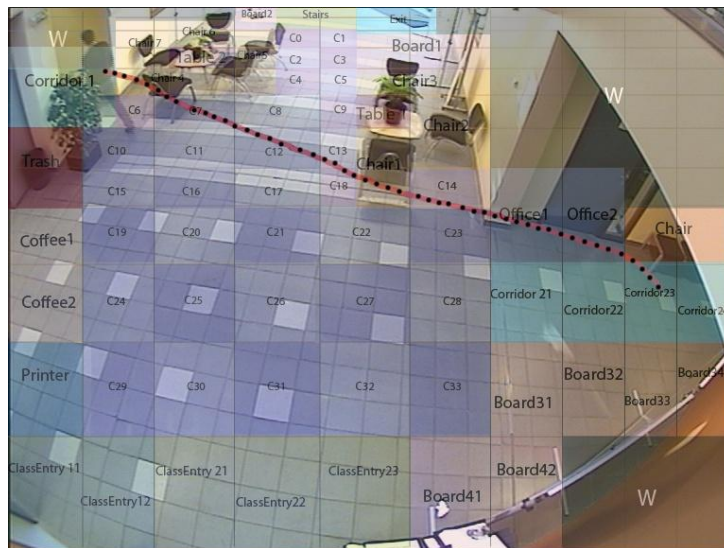


FIGURE 3.6 – Exemple de trajectoire passant par plusieurs RI

La trajectoire est donc transformée en une liste de RI correspondants aux segments en question. Pour la trajectoire précédente, nous obtenons par exemple la liste suivante : { "Corridor23" ; "Chair" ;

"Office2"; "Office1"; "C23"; "C14"; "Chair1"; "C18"; "C13"; "C12"; "C8"; "C7"; "Chair4"; "Corridor1"}.

Cette liste représente les objectifs que devra suivre, par la suite, un agent dans la simulation. En effet, chaque agent créé au sein de la simulation sera doté d'un comportement de déplacement. Ce comportement sera composé :

- d'algorithmes de déplacement¹ permettant de réaliser des planifications de chemins entre deux points et des évitements d'obstacles mobiles
- d'une liste de motivations (ou objectifs) correspondants aux RI que l'agent devra visiter durant son déplacement

Ces listes peuvent être différentes d'un agent à un autre selon les trajectoires suivies. Il est nécessaire de retenir, à ce niveau, que pour un agent dans notre simulation, une motivation, un objectif ou une RI représentent des termes équivalents. En l'occurrence, la région d'intérêt intitulée "Corridor1" est l'objectif final de l'agent justifiant le comportement de déplacement de l'individu de la figure 3.6.

Par ailleurs, l'individu peut passer un temps plus au moins court dans chaque région d'intérêt. Pour une simulation plus réaliste, nous devons tenir compte du temps passé dans chaque zone. Pour celles représentant une zone d'intérêt principale, ce temps correspondra à la durée de l'activité correspondante (par exemple attente sur la zone machine à café). Cette information temporelle découle du nombre de points relevés² dans chaque zone par laquelle passe la trajectoire et renseigne sur le niveau d'intérêt que porte une personne à la région concernée. Nous pouvons remarquer que dans la figure 3.6³ la personne ne fait que traverser le hall.

Autre exemple, un consommateur peut déambuler dans un supermarché en passant d'un rayon à un autre. Par moments, il s'attarde devant un rayon en particulier pour chercher ou récupérer un produit. Les points de sa trajectoire seraient donc espacés par moments et proches par d'autres, ce qui permet de soulever un changement de rythme.

Dans notre cas d'étude, les trajectoires étant composées de 50 coordonnées, les espaces séparant chaque paire de points sont de dimension variable. La figure 3.7 présente le déplacement d'une personne qui prend naissance au niveau du point encerclé en noir (coin supérieur gauche de l'image) et se termine en face de l'entrée d'un bureau (côté droit de l'image). Les extrémités de cette trajectoire affichent une concentration importante de points relativement à ceux situés au milieu. Nous pouvons en déduire que la personne ayant réalisé ce déplacement a passé un certain moment à côté de la chaise ou a été assise dessus, ensuite elle s'est dirigée vers le bureau et s'est arrêtée devant la porte, pour éventuellement l'ouvrir et accéder à l'intérieur. Par conséquent, dans le cas où on intègre la notion du temps accordée à chaque RI, une trajectoire est alors traduite en liste de couples (Label de la RI, Temps en nombre de points). La trajectoire de la figure 3.7 est ainsi transformée en {"C4",2), ("Chair5",9), ("C4",5), ("C8",4), ("C13",3), ("Chair1",3), ("C14",1), ("C23",3), ("Office1",5), ("Office2",5), ("Corridor22",2), ("Chair",8)}.

1. repris de la littérature correspondante - l'objectif de cette thèse n'étant pas de proposer de tels algorithmes

2. Ce nombre de points est proportionnel au nombre de trames vidéo où l'individu est dans la zone considérée et à la durée de la trajectoire totale

3. Le découpage en zones correspond au découpage affiché auparavant, mais avec une vue caméra

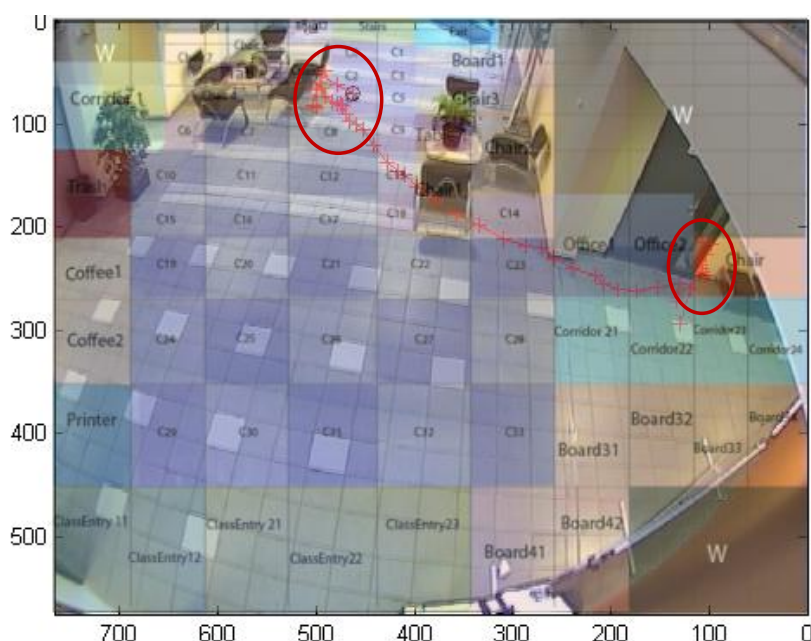


FIGURE 3.7 – Exemple de trajectoire composée de points irréguliers en présentant des concentrations au niveau des deux extrémités

La manière de gérer la liste de motivations d'un agent est relative à un des deux types de comportement suivants :

- Un comportement "statique", pour lequel un agent parcourra une liste de motivations définie à la création de l'agent (avant l'exécution de la simulation). Il représente un comportement prototype et concerne un ensemble d'agents partageant les mêmes motivations. Il existe, bien évidemment, une collection de comportements prototypes dont chacun décrit un parcours simple tel qu'un passage d'un couloir à un autre comme vu précédemment. Suite à l'achèvement des listes de motivations, au cours de la simulation, les agents terminent leur cycle de vie.
- Un comportement "dynamique", pour lequel un agent construira et parcourra incrémentalement une liste de motivations au cours de la simulation. Contrairement au comportement "statique", il est déterminé en ligne et permet des variations de comportements d'une typologie d'agent à une autre. Pour ce faire, imaginons les différentes motivations rassemblées dans une structure générale. Cette structure doit être apprise sur la base des listes de RI initiales et permet de générer différentes listes de motivations. Les listes engendrées doivent, par ailleurs, garder la cohérence des comportements et être en concordance avec la réalité. De ce fait, à la place de se limiter à un nombre fini et réduit de trajectoires, la structure apporte une diversité considérable de déplacements qui se rapprochent de la réalité et s'effectuent au cours de l'exécution de la simulation. Aussi, les agents auront plus de liberté dans leur déplacement, et de souplesse pour passer d'un comportement à un autre avant d'achever leur cycle de vie.

Avec le premier type de comportement, utilisant une seule liste de motivations et s'appuyant sur l'hypothèse qu'un agent exécute une seule trajectoire, il serait nécessaire de créer deux agents pour

reproduire un individu réalisant une combinaison de plusieurs comportements observés (par exemple, dans notre cas d'étude, un étudiant sortant d'une salle de classe pour aller fumer mais faisant un détour par la machine à café). Avec le second type de comportement, il serait possible de combiner les deux comportements et l'agent, doté d'un comportement "dynamique", effectuerait un déplacement plus élaboré. D'où l'intérêt d'explorer un moyen de regrouper tous les comportements statiques dans une seule et unique structure.

Afin de tester la faisabilité de ces deux types de comportements, nous avons appliqué, en premier lieu, une technique de segmentation, la technique de EM, engendrant des listes de motivations (voir section 2.4.4.2) spécifiques aux différents comportements observés. En deuxième lieu, nous avons appliqué les Modèles de Markov Cachés (MMC) pour tester l'utilité de s'appuyer sur une structure générale pouvant générer des comportements plus variés (voir section 3.3.2).

Avant d'aborder plus en détails la présentation des techniques d'extraction de comportements à partir d'observations, nous terminons la description de notre méthodologie de construction de simulation multi-agents en abordant l'étape de validation. Celle-ci correspond à la seconde partie de notre approche (figure 3.1) et repose sur l'utilisation de données collectives.

3.2.3 Données collectives observées pour la vérification du modèle

Sous la base des trajectoires extraites de la vidéo, une carte des points chauds et des points froids peut être construite, de façon à ce que chaque zone soit caractérisée par le nombre de fois où elle a été visitée par une trajectoire. Elle est appelée, de ce fait, la carte de chaleur du monde réel et reflète un fait conforme à un échantillon d'observations prélevé sur une période de temps. Cette carte permet alors de visualiser la répartition des zones les plus fréquentées, ainsi que les zones les plus désertes. L'information qu'elle peut prodiguer peut être considérée comme source de vérification ou de validation. Cette carte est représentée dans la figure 3.8.

Cette trace de données collectives, ainsi obtenue, va nous servir de moyen de comparaison pour évaluer la qualité du résultat de la simulation et vérifier indirectement la cohérence des comportements individuels appris. En effet, les agents ayant reproduit les déplacements prototypes, nous obtenons comme résultat une carte de points chauds et des points froids simulée. Elle est obtenue incrémentalement en augmentant de 1 un niveau de fréquentation associé à chaque cellule pour chaque agent présent exécutant sa trajectoire (liste de motivations).

Nous utilisons une métrique, appelée coefficient de corrélation [Rodgers et Nicewander, 1988], pour comparer la carte issue de la simulation (A) par rapport à la carte du monde réel (B). La formule qui calcule le coefficient de corrélation est présentée dans l'équation suivante.

$$r_{A,B} = \frac{\sum_m \sum_n [(A_{mn} - \bar{A})(B_{mn} - \bar{B})]}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)}} \quad (3.1)$$

avec m et n les nombres des lignes et des colonnes de la matrice A et de la matrice B respectivement, $\bar{A}$ et $\bar{B}$ les moyennes des éléments de A et B respectivement.

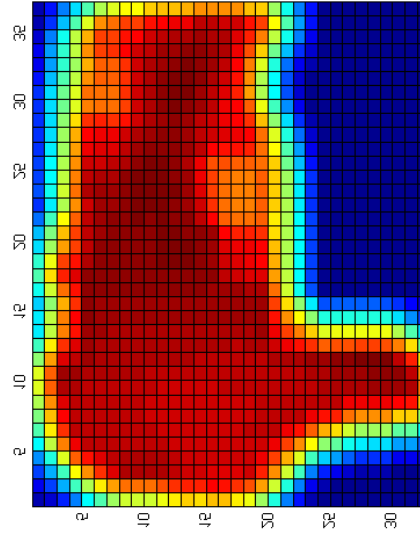


FIGURE 3.8 – Carte des points chauds et des points froids du monde réel, réalisée à partir des 682 trajectoires extraites de la vidéo

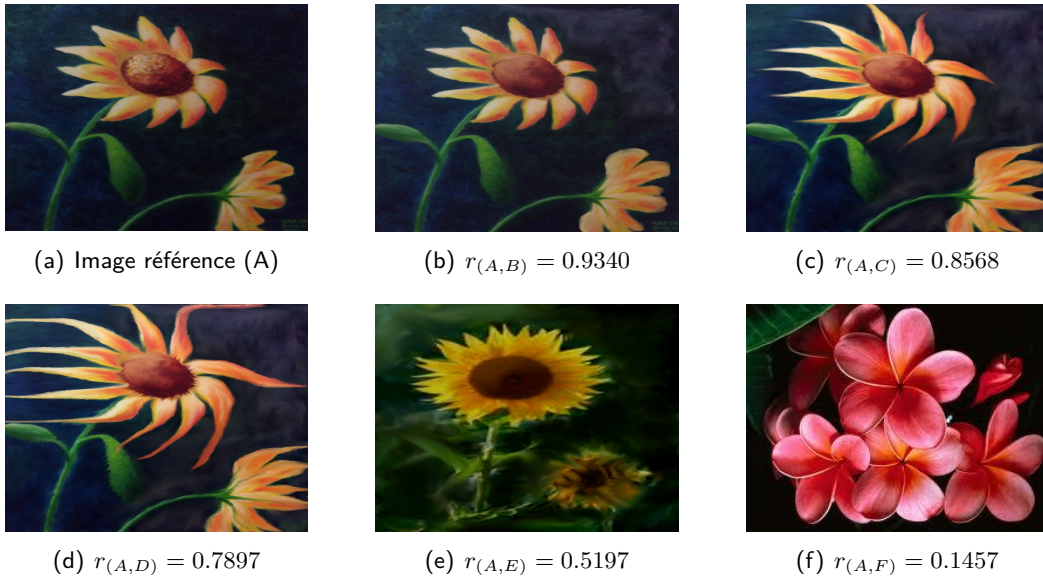


FIGURE 3.9 – Coefficients de corrélation entre les images et la première.

Ce coefficient de corrélation est une mesure symétrique caractérisant une relation linéaire positive ou négative. Plus $|r_{A,B}|$ est proche de 0, plus la relation entre A et B est inexistante ou faible ; et plus $|r_{A,B}|$ est proche de 1, plus la ressemblance entre A et B est forte. Ainsi, si A et B sont fortement semblables, le résultat de la simulation est considéré comme effectif et valide. Il est donc essentiel de fixer un seuil de corrélation à partir duquel nous jugeons que les deux matrices, de la carte réelle et de la carte résultat, sont suffisamment semblables pour valider le résultat de la simulation. La figure

3.9 illustre sur quelques images de synthèse la valeur du coefficient de corrélation (visuellement un coefficient supérieur à 0.9 correspond à une ressemblance importante).

Dans le cas où nous obtenons un coefficient de corrélation, entre la carte réelle et la carte simulée, inférieur au seuil fixé, un retour à la phase de conception de la simulation sera nécessaire, et cela, dans le but d'ajuster les paramètres d'entrée de la simulation. Il faut alors envisager différentes possibilités de remise en cause de la simulation. Le tableau 3.1 liste les différents niveaux de conception où il est possible d'intervenir, les motifs de la défaillance et, le cas échéant, les solutions proposées. La remise en cause de l'implémentation de la simulation a été mise de côté en tant que défaillance effective, pour la simple raison qu'elle est basée principalement sur des algorithmes standards et réputés dans la littérature pour leur efficacité. Pour davantage de détails et d'illustrations, des exemples concrets utilisant cette métrique seront présentés dans le quatrième chapitre.

Défaillances	Motifs	Solutions
Simulation	Non mis en cause.	
Agentification	Apprentissage non effectif.	Adapter les paramètres de la technique de clustering.
		Envisager une autre structure de modèle d'agent.
Données	Traitement d'extraction pas très fiable.	Prétraiter les données incomplètes ou bruitées.
		Améliorer le paramétrage de la technique de traitement des données.
	Nombre de données insuffisant.	Procéder à de nouvelles acquisitions de données d'apprentissage.

TABLE 3.1 – Récapitulatif des phases, mises en cause dans la méthodologie, dans le cas d'une simulation non validée

3.3 Extraction de comportements à partir d'observations

Nous avons exprimé jusqu'à présent comment, à partir d'un ensemble de trajectoires, obtenir des listes de motivations que nous injectons aux agents de la simulation initialement dépourvus d'objectifs. Nous consacrons, de ce fait, la suite de ce chapitre à la présentation des opérations permettant d'obtenir effectivement cet ensemble de trajectoires. Comme évoqué précédemment, nous déclinons notre méthodologie selon deux techniques d'apprentissage différentes. L'une qui porte sur la technique d'EM, la seconde applique les MMC. Ces deux techniques sont expérimentées afin de tester deux procédés totalement différents pour apprendre et traiter des comportements variés de déplacements d'individus.

3.3.1 Application de l'algorithme EM

Le choix de l'algorithme EM est lié au fait que celui-ci est considéré, dans la littérature, comme étant simple, efficace et capable d'estimer les modèles dans le cas de données bruitées et/ou man-

quantas. Dans notre cas, l'existence de certaines données erronées est principalement due aux limites des algorithmes de traitement vidéo en présence de problèmes d'occultation ou de séquences incomplètes.

Nous avons appliqué cet algorithme pour estimer les paramètres d'un modèle de mélange gaussien et obtenir la partition optimale des données par rapport à l'homogénéité des classes de la partition. Pour ce faire, nous avons utilisé l'outil de classification par EM sur Weka⁴, un logiciel de data mining. Disposant des 682 instances organisées en 100 attributs (voir section 3.2.1), il convient de fixer le nombre de classes que nous souhaitons obtenir afin de pouvoir appliquer le clustering. Or, pour choisir le nombre le plus approprié et sans procéder par tâtonnement, une méthode de validation croisée (plus connu sous le terme anglais : cross validation (CV)) peut être couplée à l'algorithme EM. Nous exposons brièvement la variante la plus utilisée de CV, la K-validation croisée (K-fold cross-validation), dans la section suivante.

3.3.1.1 Détermination du nombre optimal de clusters

La validation croisée est une méthode statistique utilisée pour évaluer ou comparer des algorithmes d'apprentissage. Elle permet, en l'occurrence, d'estimer la performance d'un modèle appris à partir des données disponibles en calculant la moyenne des mesures de performance des différents échantillonnages. Elle permet, également, d'utiliser les échantillons dans un test d'hypothèses statistiques pour démontrer qu'un algorithme est meilleur qu'un autre pour l'ensemble de données disponibles. L'évaluation d'un algorithme d'apprentissage est donc réalisée en divisant l'ensemble des données en deux parties. Une partie est utilisée pour l'apprentissage du modèle et une autre pour la validation du modèle. L'appellation croisée est due au fait que ce processus est réalisé selon des cycles successifs où sont échangés les ensembles d'apprentissage et de validation de telle sorte que toute donnée devrait être, à un moment donné, validée. La forme la plus simple de la validation croisée est la K-validation croisée.

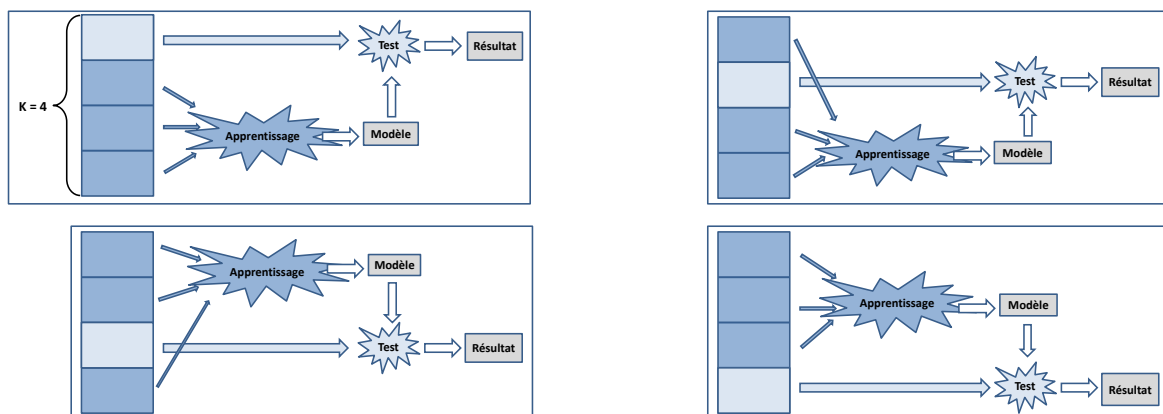


FIGURE 3.10 – Principe de la Validation Croisée avec $K = 4$

4. <http://www.cs.waikato.ac.nz/ml/weka/>

Le principe de cette méthode, illustré dans la figure 3.10, est de décider, tout d'abord, du nombre K fixe de partitions. Ensuite, les données sont réparties en K partitions approximativement égales dont une partition est utilisée pour le test, les $K-1$ restantes pour l'apprentissage. Le processus est répété K fois (par permutation circulaire) ; une mesure de performance (ou un taux d'erreur) est généralement calculée telle que l'erreur équivaut à la moyenne des erreurs sur les K partitions. En pratique, le nombre de partitions le plus utilisé est $K = 10$, il représente un bon compromis entre le nombre d'exemples pour l'apprentissage et pour le test. Cette valeur est particulièrement intéressante parce qu'elle permet des prédictions utilisant 90% des données, ce qui est très proche d'une généralisation sur l'ensemble des données. Dans la figure 3.10, K est fixé à 4 pour une question de simplification et de lisibilité.

Dans notre cas d'étude, nous appliquons cette méthode afin de déterminer le nombre de clusters optimal pour un meilleur résultat d'apprentissage. En effet, le rôle de EM est d'estimer les paramètres d'une distribution de probabilité définie suivant un ensemble de modèles gaussiens. Cependant, le nombre de ces modèles (ou groupe de données) est généralement spécifié *a priori* par le concepteur, ce qui ne garantit pas l'optimalité du résultat. Or, CV peut décider du nombre de groupes à créer. L'algorithme 4 définit ce processus.

Algorithme 4 Détermination du nombre optimal de clusters pour EM

Entrées : $n \leftarrow 1$: le nombre de clusters

$K = 10$: le nombre de partitions

X : ensemble d'apprentissage

Etape 1 X est divisé aléatoirement en K partitions.

Etape 2 EM est effectuée K fois en utilisant les K partitions de la manière habituelle de CV.

Etape 3 la logvraisemblance est moyennée sur l'ensemble des K résultats.

Etape 4 **si** la logvraisemblance a augmenté **alors**

$n \leftarrow n + 1$

retour **Etape 2** .

Sinon Terminer.

Appliquant cet algorithme sur les 682 instances trajectoires dont nous disposons, nous obtenons 24 clusters. Chaque cluster regroupe des trajectoires similaires dont la moyenne et la matrice de covariance permettent de le représenter.

3.3.1.2 Trajectoires moyennes : comportements prototypes

Les 24 clusters obtenus, suite à l'application de EM et de CV sur l'ensemble des trajectoires, correspondent à des groupes de trajectoires distinctes. Nous retenons donc, dans notre exemple, 24 trajectoires moyennes qui caractérisent des prototypes de comportements de déplacement, chacun spécifiant des données et des propriétés caractéristiques.

Plus précisément, étant donné $\{C_k\}_{1 \leq k \leq 24}$ l'ensemble des clusters obtenus. Chaque C_k est caractérisé par les informations suivantes :

- $|C_k|$: la cardinalité du cluster k ;

- $\%C_k$: le pourcentage de données d'apprentissage appartenant au cluster k ;
- $\{C_{k,i}\}_{1 \leq i \leq |C_k|}$, l'ensemble des données appartenant au cluster C_k ;
- μ_k , la moyenne du cluster k ;
- σ_k , la matrice de covariance du cluster k .

L'ensemble des composantes de la moyenne du cluster $\{\mu_k^i\}_{1 \leq i \leq 100}$ décrit donc la trajectoire moyenne de ce groupe dont les coordonnées sont $(\mu_{k,2*j+1}, \mu_{k,2*j+2})$ avec $0 \leq j \leq 49$ dans un espace 2D. Les 24 trajectoires moyennes obtenues sont tracées dans la figure 3.11.

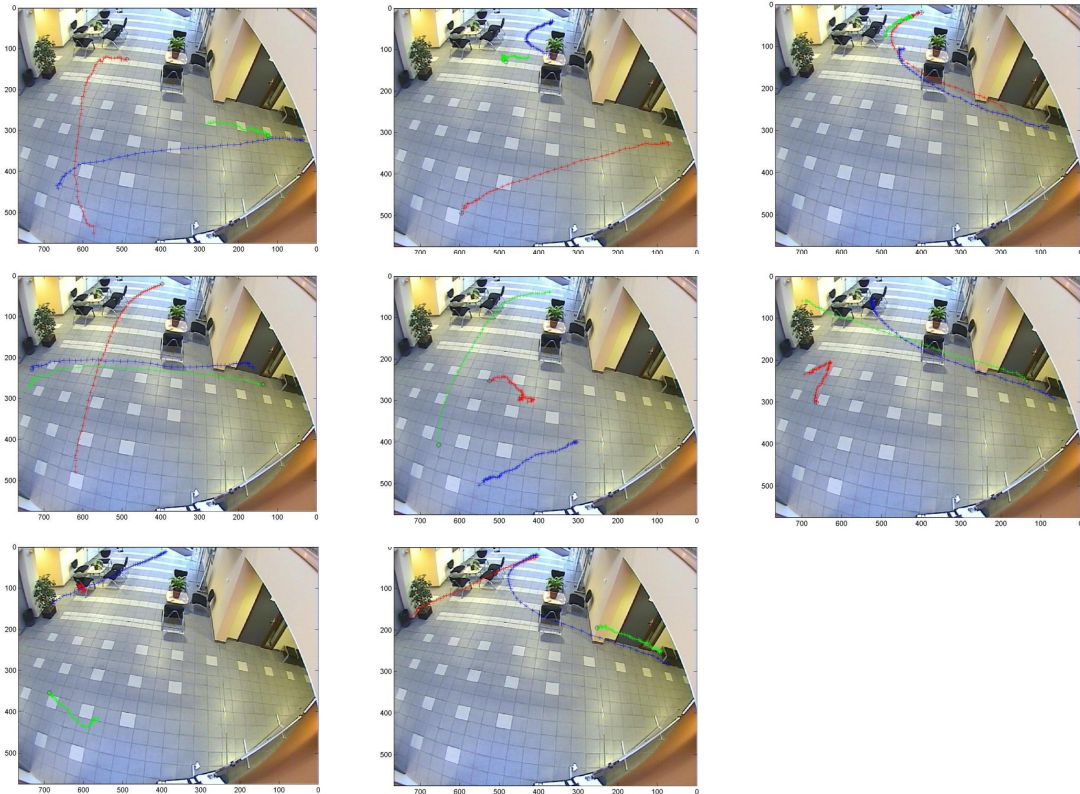


FIGURE 3.11 – Les 24 trajectoires moyennes (3 par image)

Visuellement, un grand nombre des trajectoires résultantes semblent réalistes. Elles caractérisent clairement et distinctement un passage évident au sein de l'environnement. Nous choisissons trois de ces trajectoires que nous superposons sur les données d'apprentissage appartenant aux clusters correspondants (voir figure 3.12). La figure 3.12(a) montre que les trajectoires (en bleu) regroupées dans le neuvième cluster sont particulièrement semblables et suivent toutes approximativement le même chemin. Nous pouvons constater que leur trajectoire moyenne (en rouge) suit parfaitement le même flux. A l'inverse de ce cluster, ceux retracés dans les figures 3.12(b) et 3.12(c) montrent une dispersion plus importante. Toutefois, la tendance générale est intelligible et nous la retrouvons effectivement par la trajectoire moyenne. Au sein d'un même groupe, les trajectoires qui paraissent

aberrantes à ce stade ne le sont pas forcément. En réalité, elles se présentent comme étant des trajectoires inhabituelles et isolées par rapport aux autres, réalisées par un nombre minime de personnes, et par conséquent, intégrées dans les clusters les plus proches. Il serait, bien évidemment, moins approprié de créer un nouveau cluster contenant une seule et unique trajectoire.

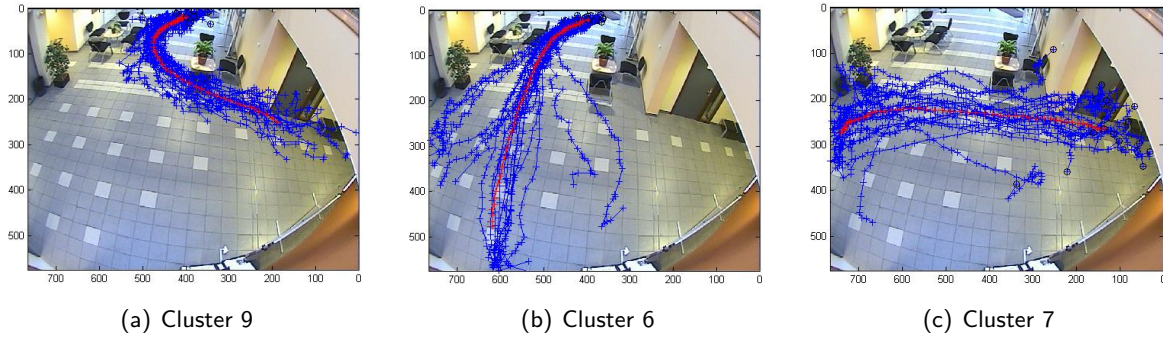


FIGURE 3.12 – Exemple de trajectoires moyennes complètes, superposées sur l'ensemble des instances des clusters correspondant

Par ailleurs, il existe des trajectoires moyennes qui prennent naissance ou disparaissent au centre de l'environnement, comme nous pouvons le remarquer dans la figure 3.11. Deux de ces trajectoires incomplètes sont détaillées dans ce qui suit.

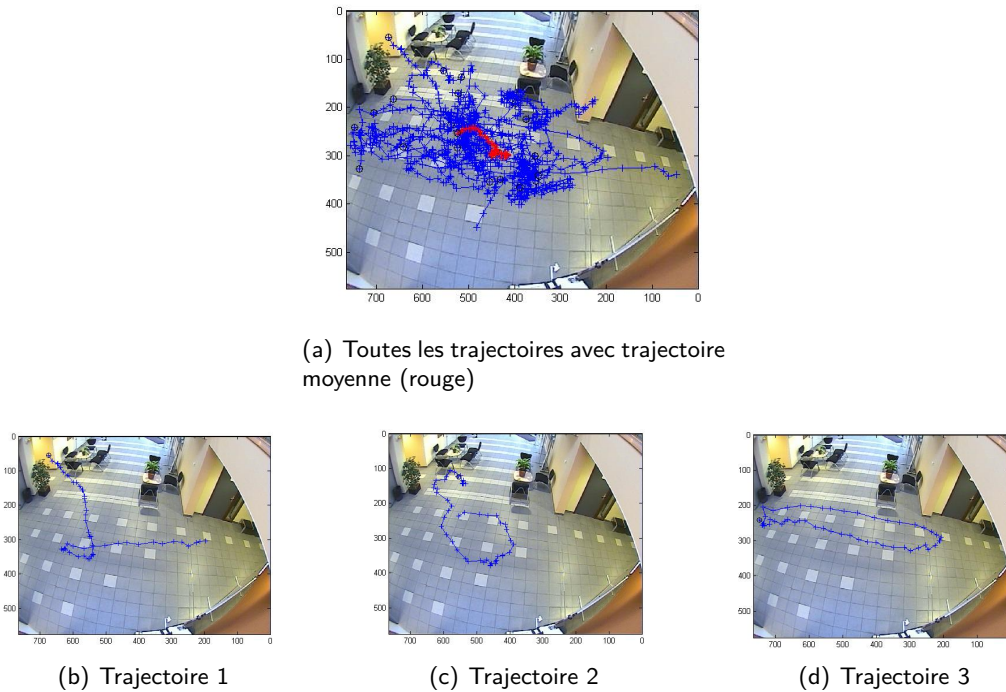


FIGURE 3.13 – Trajectoires du cluster 18

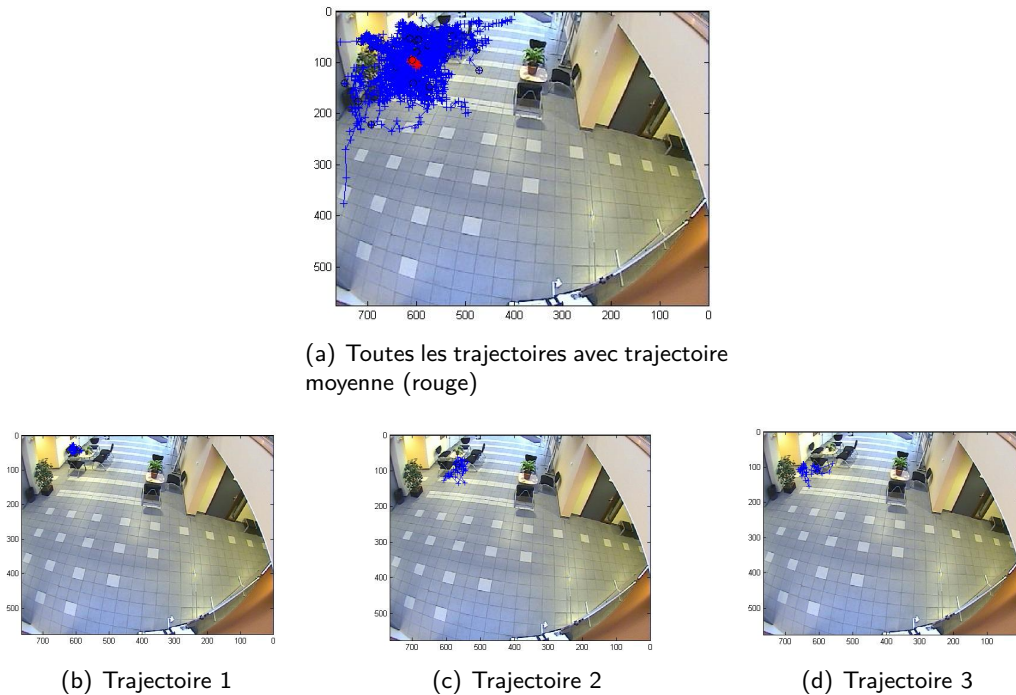


FIGURE 3.14 – Trajectoires du cluster 12

Ainsi, la figure 3.13(a) montre une concentration de trajectoires au centre de l'environnement. Les formes de ces trajectoires indiquent des comportements de déplacement singuliers. En effet, elles révèlent des comportements qui ne marquent pas simplement un passage à travers l'environnement, mais des comportements plus complexes. Trois de ces trajectoires ont également été isolées comme le montrent les figures 3.13(b), 3.13(c) et 3.13(d). Ces déplacements hors normes peuvent être réalisés, par exemple, par une personne se promenant dans le hall discutant au téléphone, ou en attendant le café faisant alors un aller-retour aux alentours, etc. Ainsi, le cluster concerné (cluster 18) n'est pas dépourvu d'intérêt et son existence illustre une région très fréquentée.

Aussi, une autre trajectoire moyenne qui se distingue du reste est illustrée dans la figure 3.14. Les trois trajectoires, représentées individuellement, représentent des déplacements qui traversent la zone de détente et, particulièrement, des mouvements de personnes assises. Ce qui explique la concentration de points au niveau des chaises. Cette information est aussi pertinente que le reste des trajectoires. Elle permet, effectivement, de constater qu'un certain nombre d'individus a tendance à choisir cet emplacement en particulier.

Ayant à disposition tous les résultats concernant le nombre de clusters optimal, la répartition dans chaque cluster des trajectoires initiales et la connaissance des trajectoires moyennes, il est utile de s'assurer de la proximité des trajectoires au sein d'un même cluster et leur éloignement s'ils appartiennent à deux clusters différents. Ceci permet de vérifier la qualité du résultat du clustering et explorer sa capacité à isoler des groupes distincts de trajectoires au moyen d'une méthode évaluant la proximité des données entre elles. La méthode des dendrogrammes, exposée dans la section suivante,

semble être une bonne candidate pour procéder à cette vérification.

3.3.1.3 Vérification par classification ascendante hiérarchique

La classification hiérarchique est une méthode de segmentation fournissant une hiérarchie de partitions classifiant des individus ayant des caractéristiques similaires. On distingue deux approches différentes : l'approche ascendante et l'approche descendante. Nous nous intéressons dans ce qui suit à la première approche qui part du principe que chaque trajectoire représente un individu formant sa propre classe singulière. Cette approche nous sert de moyen de comparaison et de vérification en confrontant les résultats de EM. Cette vérification n'est pas nécessairement abordée à chaque fois qu'on utilise la technique de EM. Néanmoins, dans notre étude elle aura pour effet d'appuyer la pertinence et la cohérence de l'hypothèse du choix de EM ou bien de le remettre en cause. Pour ce faire, nous allons au préalable exposer le fonctionnement et l'application de cette méthode.

Partant de l'ensemble de toutes les données représentées chacune par une classe, la classification ascendante hiérarchique a pour objectif d'agréger deux à deux les classes les plus proches (C_a, C_b) au sens d'une certaine mesure de similarité. L'algorithme 5 décrit cette approche.

Algorithm 5 L'algorithme de classification hiérarchique ascendante

Entrées : $\chi^{(0)} = x_1, \dots, x_n$,
 $d(C_a, C_b)$: une mesure de similarité entre C_a et C_b ,
 $P_0 = x_1, \dots, x_n$: la partition initiale.
Initialiser $d_{min} \gg 0$
pour $i = 1$ à $n-1$ **faire**

pour chaque couple (C_a, C_b) de P_{i-1} **faire**
si $d(C_a, C_b) < d_{min}$ **alors**
 $C_{opt} = (C_a, C_b)$
 $d_{min} := d(C_a, C_b)$
fin pour chaque
 $C_{fusion} = C_a \cup C_b$
 $P_i = (P_{i-1} - \{C_a, C_b\}) \cup C_{fusion}$
fin pour

Parmi les méthodes les plus souvent utilisées pour définir la métrique de mesure de similarité ou dissimilarité, nous trouvons :

- 'Single' : représentant la distance la plus petite entre deux individus choisis parmi C_a et C_b :

$$d(C_a, C_b) = \min\{d(x, y)\} , x \in C_a \text{ et } y \in C_b \quad (3.2)$$

- 'Complete' : représentant la distance la plus grande entre deux individus choisis parmi C_a et C_b :

$$d(C_a, C_b) = \max\{d(x, y)\} , x \in C_a \text{ et } y \in C_b \quad (3.3)$$

- 'Average' : représentant la distance moyenne entre les individus des deux groupes :

$$d(C_a, C_b) = \frac{\sum_{i=1}^{n_a} \sum_{j=1}^{n_b} d(x_i, y_j)}{n_a * n_b}, \quad x_i \in C_a \text{ et } y_j \in C_b \quad (3.4)$$

Etant munis d'une métrique (mesure de similarité telle que la distance euclidienne) et d'un critère d'agrégation, nous obtenons alors une hiérarchie de partitions représentée par un arbre définissant la structure de classification (dendrogramme).

A partir des données dont nous disposons, nous avons créé l'arbre hiérarchique selon l'algorithme 5 avec la distance moyenne ("Average") en tant que mesure de similarité.

Le choix du nombre de classes est déterminé *a posteriori* en coupant la hiérarchie à un certain niveau. Plusieurs critères peuvent intervenir pour décider à quel niveau le découpage doit être réalisé pour obtenir une bonne partition. En ce qui concerne notre objectif, rappelons-nous, nous cherchons à utiliser cette méthode de classification ascendante hiérarchique afin de vérifier la pertinence du résultat de EM. Pour cela, nous décidons de réaliser un découpage de telle sorte que nous obtenions le même nombre de clusters que celui obtenu avec EM, en l'occurrence 24 clusters. La figure 3.15 illustre le dendrogramme obtenu.

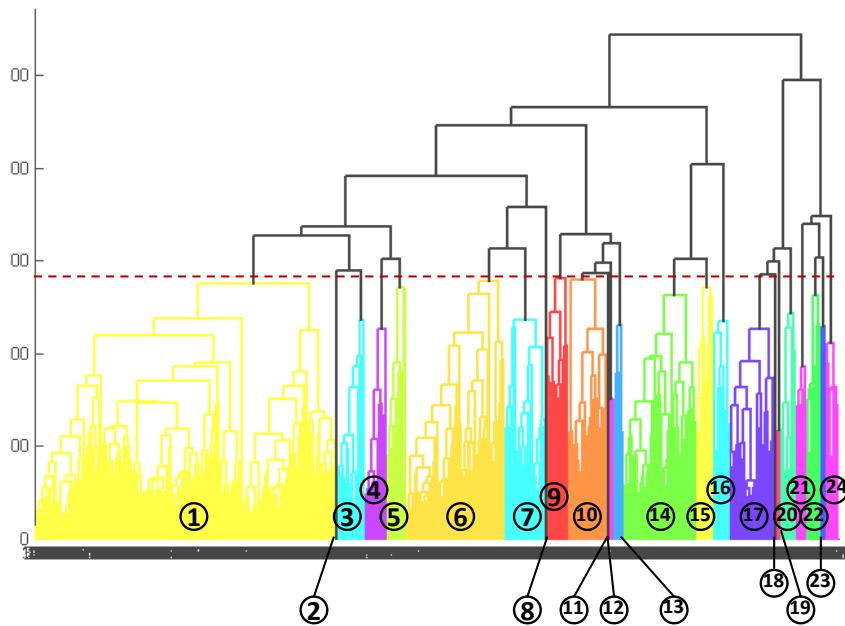


FIGURE 3.15 – Coupe transversale du dendrogramme entraînant 24 segments

Les couleurs attribuées aux 24 sous-arbres représentent des classes contenant les données disposant d'une similarité proportionnelle au niveau de coupe réalisée. Nous avons affecté des numéros à chacun des groupes pour une question de commodité. Nous remarquons que la distribution des instances est moins équilibrée qu'avec l'algorithme EM. En effet, il existe des classes singulières (2, 8, 12 et 18) contenant une seule donnée alors qu'avec EM, le plus petit cluster contient 11 éléments et

représente 2% de la totalité des données. Il s'avère que les trajectoires représentées par ces classes singulières réalisent des déplacements bidirectionnels dans le sens où des aller-retour, entrepris par la même personne, ont été détectés. A l'opposé, certaines classes obtenues par le dendrogramme correspondent fortement aux clusters estimés par l'algorithme EM, notamment les sous-arbres 16, 19, 21, 22 dont 100% des données appartiennent aux cluster 4, cluster 4, cluster 3, cluster 21 respectivement. Le cluster 4 englobe donc les deux sous-arbres 16 et 19 dont quatre exemples sont présentés dans la figure 3.16(a) où les trajectoires en bleu appartiennent au groupe 4 et au sous-arbre 16, et les trajectoires en rouge appartiennent au groupe 4 et au sous-arbre 19. Cependant, le sous-arbre 19 est pauvre en données et ne contient que quatre trajectoires.

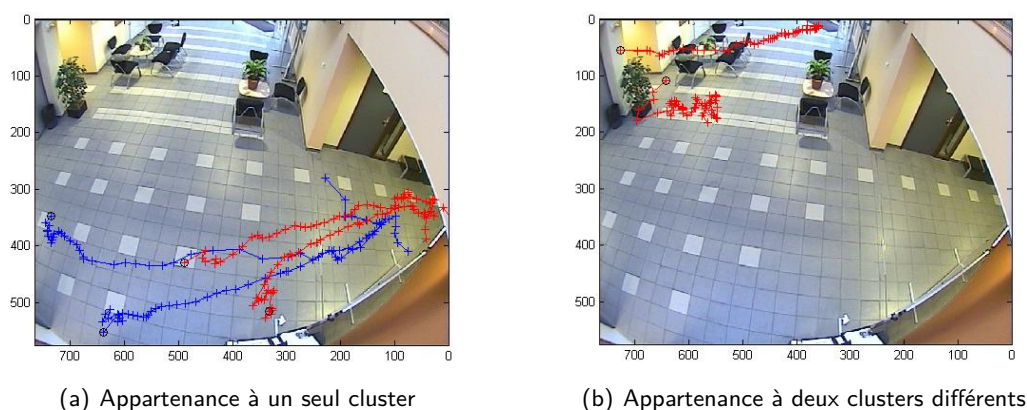


FIGURE 3.16 – Exemple d'instances

Par ailleurs, le sous-arbre 1 est particulièrement large et contient plus de 280 données appartenant à différents clusters avec EM. Deux de ces trajectoires ont été reportées sur la figure 3.16(b), l'une appartenant au cluster 15 et l'autre au cluster 13. En conclusion, l'algorithme EM permet d'obtenir de meilleurs partitions et évite l'apparition de groupes très déséquilibrés.

Suite à la vérification de la qualité et la cohérence des résultats de EM avec la technique des dendrogrammes, il est avéré qu'effectivement, la catégorisation des données initiales est appropriée pour l'étude des similarités de trajectoires. Nous sommes donc capables d'extraire des groupes homogènes de trajectoires en appliquant la méthode EM.

Les résultats du clustering étant vérifiés et validés, les trajectoires moyennes obtenues peuvent être exploitées pour construire les motivations d'agents.

3.3.1.4 Traduction des trajectoires en motivations

Jusqu'alors, nous avons appliqué la technique de EM sur l'ensemble des trajectoires dont nous disposons. Nous avons ainsi obtenu une partition de données qui paraît très satisfaisante suite à des vérifications visuelles et par la technique de classification hiérarchique. Les clusters obtenus étant assimilés à des prototypes de déplacement décrits par des trajectoires moyennes et ayant connaissance de l'environnement caractérisé par des RI, il ne reste plus qu'à traduire ces trajectoires en termes de motivations d'agents (voir section 3.2.2.2). Ces agents seront conçus pour reproduire en simulation le monde observé.

Ensuite, nous organisons les résultats de EM traduits en motivations dans un fichier. Celui-là est structuré de telle sorte que chaque ligne définisse un cluster (prototype d'un agent type) avec l'ensemble des éléments essentiels permettant de définir le nombre et le comportement de déplacement des agents de ce type. Un échantillon de ce fichier, composé de six prototypes d'agent, est présenté dans le tableau 3.2 où :

- la première colonne représente l'identifiant du type d'agent, en l'occurrence le numéro du cluster correspondant.
- la seconde colonne représente le pourcentage d'individus de même type par rapport à la population entière.
- la dernière colonne représente la liste des motivations (les RI successives) dont chacune est caractérisée par un temps représentant la durée que l'agent doit passé dans la zone d'intérêt correspondante.

Les six lignes dans le tableau 3.2 correspondent à six comportements de déplacement d'agent. Le tableau est donc utilisé comme paramètre d'entrée de la simulation pour créer la population d'agents.

Type	%	{(Motivation, Temps)}
T1	2	(C12,5),(C8,5),(C7,4),(C11,6),(C16,4),(C20,5),(C25,3),(C24,2),(C29,5),(C30,1),(ClassEntry12,1),(ClassEntry21,9)
T3	2	(Corridor24,5),(Corridor23,7),(Corridor22,5),(Corridor21,4),(C28,4),(C27,4),(C26, 1),(C31,4),(C30,5),(C29,11)
T4	3	(ClassEntry21,10),(C30,2),(C31,7),(C32,5),(C33,6),(Board31,1),(Corridor21,3),(Corridor22,7),(Corridor23,9)
T6	6	(Table1,6),(C9,3),(C5,12),(C3,10),(Board,19)
T8	3	(Office2,5),(Office1,6),(C23,6),(C22,5),(C21,5),(C20,5),(C19,6),(Coffee1,11),(Coffee2,1)
T15	7	(Corridor1,1),(C6,4),(C7,4),(Chair4,5),(Chair5,6),(C2,2),(C0,8),(C1,6),(Stairs,6),(Exit,5)

TABLE 3.2 – Extrait de l'organisation du fichier de motivations

Sur la base du fichier obtenu, dont le tableau 3.2 représente un extrait, la simulation reproduisant l'environnement étudié va créer 682 agents (le nombre initial des données d'apprentissage) dont 2% sont du type 1 (Cluster 1) exécutant la liste des motivations correspondantes, 4% sont du type 2 (Cluster 2), 2% sont du type 3 (Cluster 3), 3% sont du type 4 (Cluster 4) et ainsi de suite jusqu'au type 24 qui représente le dernier prototype. Par conséquent, chaque type d'agents créé exécute approximativement le même comportement. Ce qui revient à dire qu'avec EM nous avons extrait 24 types d'agent où chacun est conçu avec une structure fixe et unique de motivations. Dans ce cas de figure, les agents dans la simulation possèdent un cycle de vie déterministe. Il est à noter, par conséquent, que les comportements manquent de flexibilité dans le sens où il est difficile de changer de comportement ou de basculer vers un autre comportement. En effet, supposons que nous ayons par exemple un agent "buveur de café" qui se déplace de l'entrée des classes à la machine à café, et un autre agent "fumeur" qui se déplace de la machine à café à la sortie. Avec EM, nous avons deux prototypes et la simulation produit deux types d'agents différents dont le premier exécute un déplacement vers la machine à café et le deuxième exécute un déplacement depuis la machine vers la sortie. Dans le but de simuler des fumeurs qui sont aussi buveurs de café, il serait intéressant de pouvoir mêler les deux comportements. Ainsi au lieu de créer à chaque fois deux agents exécutant successivement deux comportements, un seul agent serait créé afin d'accomplir plusieurs comportements.

Pour ce faire, il est nécessaire de chercher un moyen pour créer une structure générale permettant de reproduire plusieurs comportements d'agents. Cela nous a amené à explorer l'idée d'utiliser les MMC. En effet, cette technique permet de construire une structure complète apprise à partir d'un ensemble de listes de RI différentes, en l'occurrence les listes de motivations de toutes les trajectoires initiales extraites de la vidéo. La structure obtenue peut alors être utilisée, *a posteriori*, pour générer des comportements de déplacement plus flexibles et complexes que ceux obtenus suite à l'application de l'algorithme EM.

La section suivante est consacrée à cette étude et à une application des MMC sur nos données.

3.3.2 Application des MMC

Nous avons présenté, dans le chapitre 2, ce qu'est un MMC, comment il fonctionne et quelle est sa capacité à construire des modèles caractérisant un comportement donné. Bien qu'il soit majoritairement utilisé dans des problèmes de reconnaissance, nous souhaitons l'employer afin de construire une structure unique et générale d'agent permettant de reproduire différents comportements.

3.3.2.1 Le modèle de la structure générique

Comme présenté dans le chapitre précédent, un MMC peut être assimilé à une structure de graphe composée de noeuds et d'arcs. Les noeuds représentent des états cachés où chacun couvre plusieurs observations. Les noeuds sont donc non-observables *a priori* d'où le qualificatif "caché". Une matrice d'observations nous permet de retrouver la distribution de probabilité de toutes les observations (dans notre cas, les régions d'intérêt) associées à chaque état. Les arcs représentent la possibilité de passer d'un état caché à un autre, également définie selon une loi de probabilité propre à chaque transition. Pour chaque état caché, une matrice de transitions nous permet de retrouver la distribution des probabilités de transitions vers tous les états cachés. Pour initialiser la structure, il existe un vecteur de distribution de probabilité d'états cachés initiaux permettant de déterminer le noeud de départ. Une fois les distributions des probabilités apprises à partir d'un échantillon de données d'observations (voir chapitre 2), il est possible de parcourir le graphe obtenu à partir de l'état caché initial. A chaque état caché visité, une observation est générée selon une distribution de probabilité spécifique à chaque état.

Dans notre cas d'étude, les observations représentent les RI de l'environnement. Si nous parcourons dix états cachés du graphe du MMC, nous obtenons dix observations donc dix RI formant un comportement de déplacement que doit exécuter un agent dans la simulation.

Habituellement, les méthodes de reconnaissance utilisent les modèles non-ergodiques⁵, tels que les modèles gauche-droite dans le traitement de la parole, afin d'imposer un ordre temporel au MMC. A cet effet, les observations suivent une logique d'apparition plutôt cadrée. La matrice de transitions d'états a alors une forme triangulaire supérieure. Dans notre cas, le modèle générique a pour but de reproduire plusieurs comportements différents, selon un MMC avec une matrice de transition d'états complète, représentant un modèle ergodique⁶.

5. graphe partiellement connecté (matrice de transition diagonale par exemple)

6. graphe complètement connecté (tous les noeuds sont connectés entre eux)

Partant de nos données correspondant aux trajectoires initiales issues de la vidéo, nous procédons immédiatement, contrairement à EM, à leurs traductions en listes de RI parcourus (selon la segmentation de l'environnement réalisée auparavant). Notre base d'apprentissage est composée, donc, de 682 listes de RI. L'ensemble des RI obtenus représente, de ce fait, les observations du MMC. Ce dernier apprend, par la suite, la structure du modèle général, notamment la matrice *a priori* π , la matrice d'observations B et la matrice de transitions A , à partir de ces séquences de RI (voir section 2.4.4.2). Etant donné la remarque précédente qui consiste à vouloir réaliser un modèle pouvant reproduire différents comportements sans dépendance particulière entre eux, la matrice de transition est donc complète et les transitions peuvent être établies à partir de n'importe quel état vers un autre. Ainsi, à partir de la structure apprise, on est capable de générer diverses séquences de motivations (RI), destinées aux agents de la simulation, basées sur des comportements originaux.

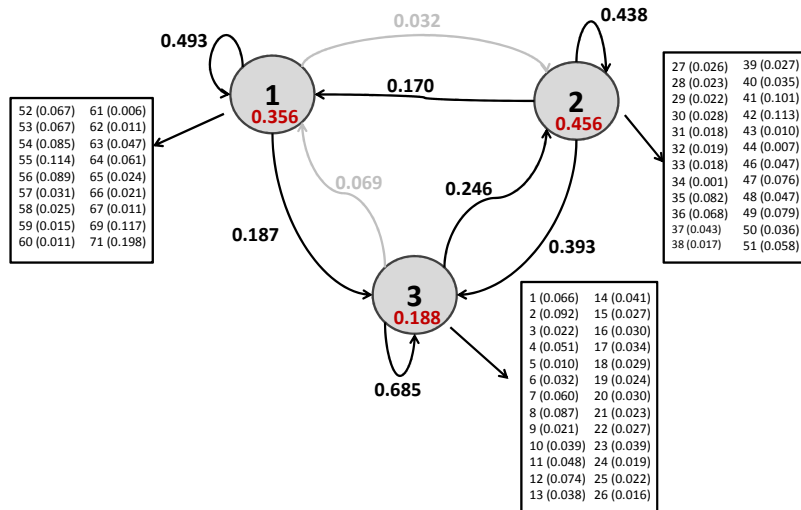
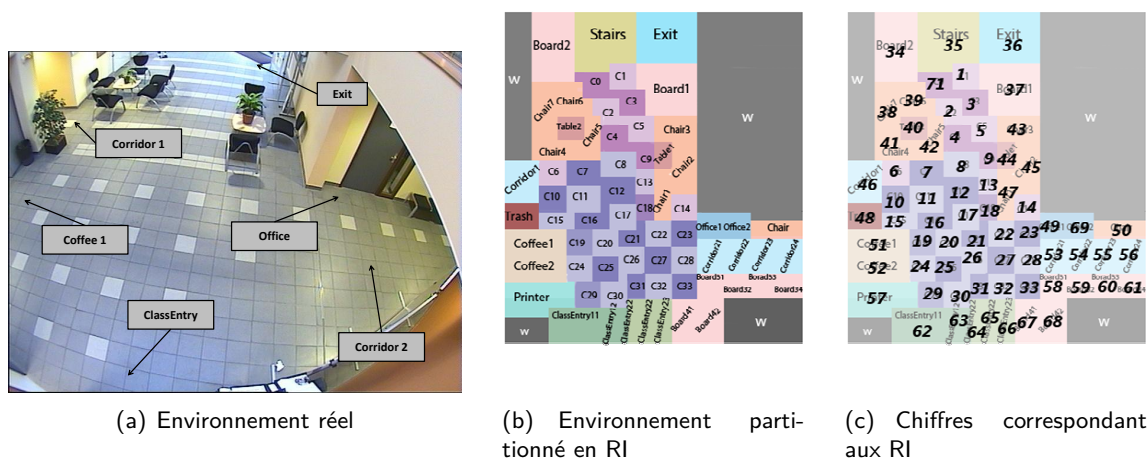
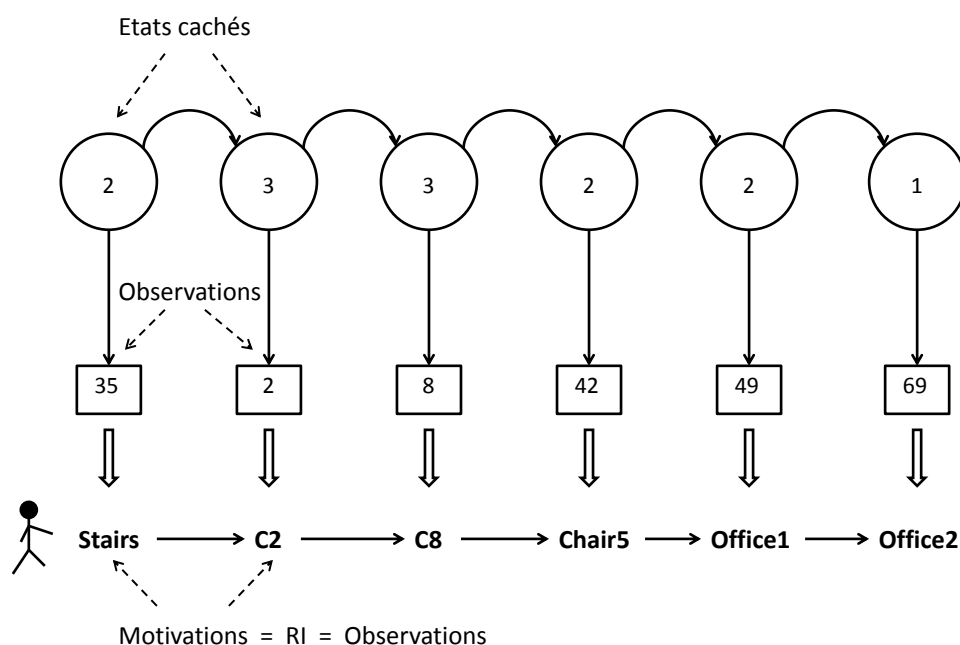


FIGURE 3.17 – Exemple d'un MMC à 3 états

La figure 3.17 représente un MMC à trois états construit à partir de nos séquences d'observations où les étiquettes des régions sont décrites par des nombres tels que présentés sur la figure 3.18. Les chiffres en rouge à l'intérieur des états de la figure 3.17 représentent la distribution *a priori* π . Dans ce modèle, nous avons une plus forte probabilité de commencer par l'état numéro 2 pour générer la première observation. En outre, les chiffres sur les transitions représentent les composantes de la matrice de transition. Enfin, les tableaux se situant à côté de chaque état représentent les observations associées (les RI) pour chacun d'entre eux et où les nombres correspondent aux cellules de l'environnement suivies par leurs probabilités entre parenthèses. Toutes ces distributions de probabilités sont estimées à partir des séquences d'observation décrivant les différentes trajectoires.



A partir de cet MMC, on est capable de générer des futurs comportements d'agents. La figure 3.19 représente un exemple de listes d'observations (correspondant aux RI de l'environnement et aux futures motivations des agents) engendrées par six états cachés parcourus à partir de l'état initial numéro 2.



Le MMC à trois états cachés obtenu est un exemple de modèle trivial inapproprié à notre étude. En effet, au sein d'un même état caché on retrouve des probabilités d'observations très proches les unes des autres, sachant qu'elles représentent des RI très éloignées dans l'environnement étudié. Cela rend la génération d'observations, dans certains cas, non consistante.

Choisir le nombre approprié d'états cachés dans un MMC, qui est généralement fixé *a priori*, n'est pas une tâche facile. Il est souvent défini de manière empirique bien qu'il puisse être décisif en termes de qualité de l'apprentissage. Pour résoudre ce problème, peu de méthodes ont été proposées [Randane *et al.*, 2003] [T.Brouad *et al.*, 1997].

3.3.2.2 Choix du nombre d'états cachés optimal du MMC

Dans notre approche, nous cherchons à fixer le nombre d'états cachés optimal de façon itérative, en calculant le diamètre du premier ordre dans chaque état. Pour plus de clarté, ayant un environnement, nous supposons que nous disposons de N observations ($N = 71$ RI dans notre cas d'étude), nous apprenons des MMC de taille différente : de 3 états jusqu'à k états, $k \prec N$. Pour chaque MMC à x états où $3 \prec x \prec k$, on calcule la dispersion de chaque état, en termes de distance entre les RI, à l'aide de l'équation du diamètre du premier ordre décrite ci-dessous :

$$D(E) = \frac{2}{n(n-1)} \sum_{i,j}^n \sqrt{(x_{o_i} - x_{o_j})^2 + (y_{o_i} - y_{o_j})^2} \quad (3.5)$$

Où E est un état, (x_{o_i}, y_{o_i}) et (x_{o_j}, y_{o_j}) représentent les coordonnées des centres des RI o_i et o_j consécutivement dans l'environnement, et n est le nombre de RI dans cet état.

Plus D est petit, plus faible est la dispersion des observations au sein d'un même état. A titre d'exemple, la figure 3.20 montre deux illustrations de l'environnement étudié dans lesquelles nous entourons des zones, contenant plusieurs RI et caractérisant différents états cachés. Nous nous contentons de représenter deux états cachés sur l'un et trois états cachés sur l'autre pour une raison de clarté. La figure de droite avec un D faible contient principalement des états aux RI juxtaposées qui pourraient décrire une zone caractéristique (Etat 1 ~ L'entrée principale du département) et donc l'état correspondant pourrait acquérir un sens interprétable (état d'entrée) sans perdre, pour autant, sa nature cachée (il englobe plusieurs observations). Par conséquent, un passage de l'état 1 à l'état 2, dans le MMC, représenterait un déplacement de l'entrée au centre-nord de l'environnement pour n'importe quelles observations générées. Par contraste, la figure de gauche montre que les états disposent d'observations assez éloignées. Ces états cachés semblent plus difficile à interpréter sauf à considérer une importante schématisation de l'environnement.

Ainsi, il est naturel de constater que plus le nombre d'états augmente, plus D est petit. Une condition d'arrêt est donc nécessaire. En conséquence, le choix du nombre optimal d'états est défini selon le meilleur compromis. Ce compromis est défini à partir des critères suivants : le plus petit diamètre du premier ordre ($Min(D)$) recueilli par un état caché d'un MMC donné ; le plus grand diamètre du premier ordre ($Max(D)$) recueilli par un état caché du même MMC ; et la moyenne de tous les diamètres du premier ordre ($Moy(D)$) recueillis par les états cachés de ce MMC. Pour obtenir le meilleur compromis, il faut maximiser $Min(D)$, minimiser $Max(D)$ et minimiser $Moy(D)$, ceci pour une succession de MMCs dont le nombre d'états cachés est augmenté de 1 à chaque itération.

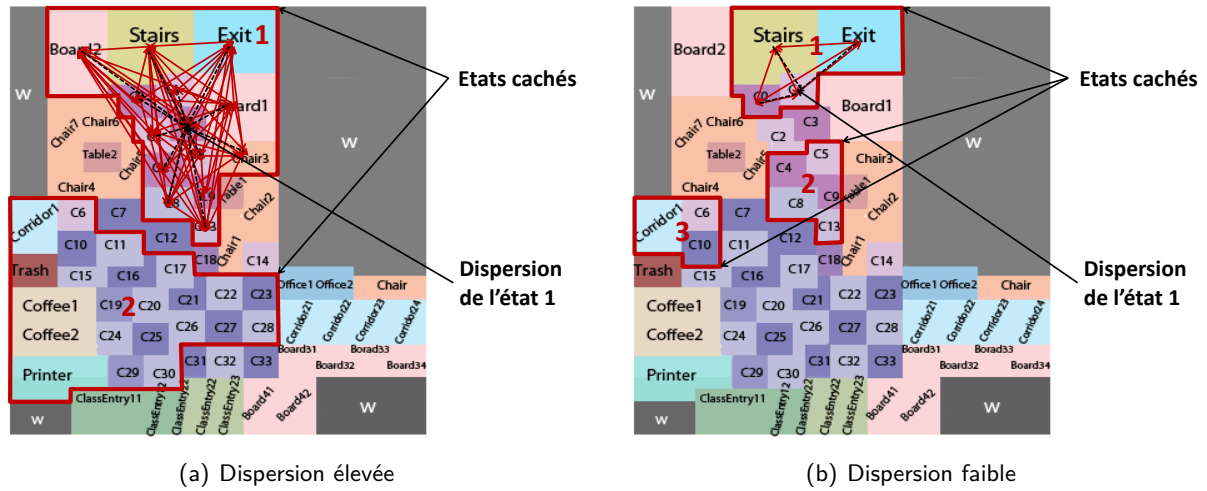


FIGURE 3.20 – Dispersion de l'état

Dans notre exemple, 28 MMC sont construits à partir de $k = 3$ états jusqu'à $k = 30$ états. Pour chaque MMC à x états, la matrice de transitions A , la matrice d'observations B et la distribution initiale des états π sont apprises. Par ailleurs, le diamètre de premier ordre $D(E)$ est calculé pour chaque état à chaque MMC. Le tableau 3.3 montre un résumé des MMC traités allant de 13 à 29 états. Les autres modèles ne figurant pas dans ce tableau, car ils n'ont pas d'incidence sur l'interprétation du résultat. Nous y rapportons la moyenne des diamètres, ainsi que le plus petit et le plus grand. La figure 3.21 représente les tendances des trois courbes des diamètres de premier ordre telles que la courbe rouge décrit les diamètres maximaux, la courbe verte décrit les diamètres minimaux et la courbe bleue décrit la moyenne des diamètres dans chaque MMC.

Nb d'états	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Moy(D)	442,90	345,07	228,20	319,64	240,76	241,61	203,04	149,47	175,95	180,91	132,94	136,36	88,72	87,69	87,26	75,82	78,93
Min(D)	142,77	20,45	9,74	39,94	0,00	2,24	2,92	21,85	4,84	1,41	1,89	4,30	0,71	0,00	0,00	0,00	0,00
Max(D)	4950,67	1145,60	1601,81	800,45	909,66	909,66	566,90	462,61	710,59	1219,02	1448,51	839,27	940,06	270,32	357,41	404,56	659,22

TABLE 3.3 – La moyenne, le minimum et le maximum du diamètre d'ordre 1 pour différents MMC à x -états

Nous remarquons que les résultats tendent à diminuer, en général, selon le nombre d'états. Nous estimons les MMC en augmentant le nombre d'états jusqu'à atteindre une continuité de $\min(D)$ égale à 0 (les MMC de 26 à 29 états). Les MMC avec au moins un $\min(D(E)) = 0$ sont considérés comme non pertinents. Le meilleur compromis dans les résultats obtenus est le MMC avec 20 états cachés. En effet, son diamètre maximum est le plus faible et le diamètre minimum est éloigné de 0.

Ainsi le MMC retenu est composé de 20 états cachés dont chacun correspond à une zone représentative de notre environnement étudié (voir figure 3.22). Il est nécessaire, cependant, de fixer le nombre d'états à parcourir dans le MMC pour générer une séquence d'observations relative à un

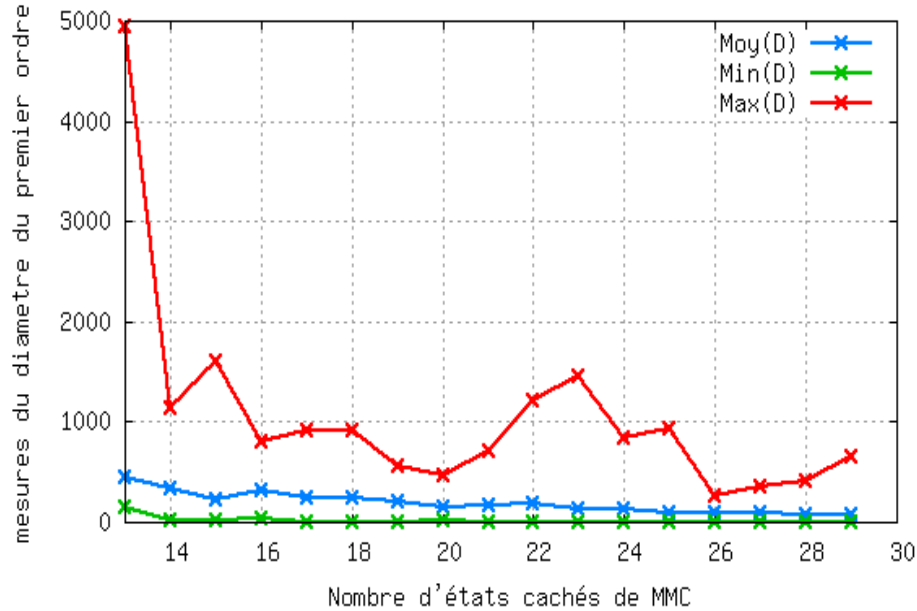


FIGURE 3.21 – Diamètres de premier ordre en fonction du nombre d'états cachés

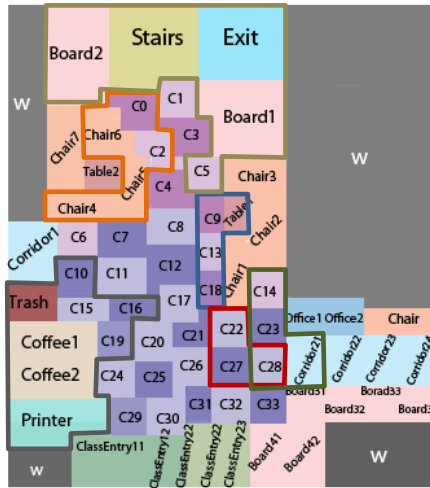


FIGURE 3.22 – Zones correspondantes à quelques états cachés extraits du MMC à 20 états cachés

déplacement convenable. Nous avons, par conséquent, fixé ce nombre à 5 observations, après avoir examiné le comportement des agents dans la simulation avec des séquences de différentes longueurs, afin d'obtenir des déplacements globalement entiers et plus proches de la réalité. Nous avons généré, par exemple, la liste d'observations suivante : (Board41, Office1, Corridor24, Corridor22, Office2) illustrée dans la figure 3.23.

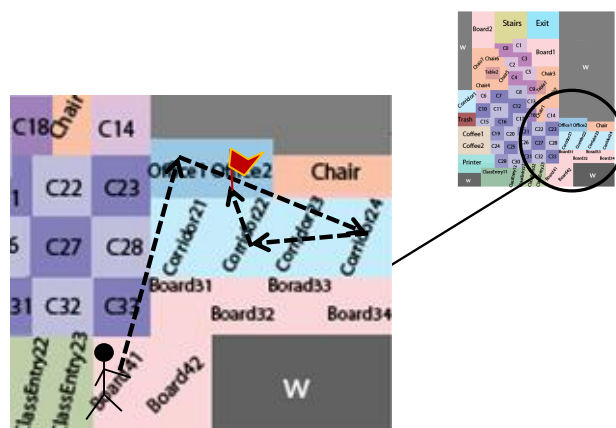


FIGURE 3.23 – Exemple de déplacement d'un agent dans la simulation suivant une liste de motivations correspondantes aux 5 observations obtenues par un MMC à 20 états

Etant donné que les séquences d'observations, destinées aux agents de la simulation, sont engendrées à partir du MMC, elles peuvent formuler plusieurs comportements de déplacement totalement distincts. Contrairement aux résultats de EM où les agents sont distribués selon des typologies données, la répartition des agents, au sein de la simulation, issue de la structure du MMC est tout à fait libre. Il existe donc autant de types d'agent que de séquences générées. Par ailleurs, le MMC ne tient pas compte de la notion du temps. Toutefois, les séquences de RI engendrées représentent uniquement les listes de motivations des agents. Par conséquent, si un individu s'arrête au niveau de la machine à café par exemple, le modèle MMC obtenu ne sera pas capable de mettre en évidence ce comportement (contrairement au modèle EM).

De plus, nous notons bien que l'attitude des agents, dont les motivations sont issues du modèle MMC, n'est pas toujours très réaliste, notamment à cause de certains déplacements incohérents (aller-retours peu réalistes). De plus, certains déplacements s'achèvent au milieu de l'environnement, un phénomène que nous avons relevé également dans les résultats du modèle obtenu avec EM. Nous déduisons, au final, que les comportements des agents au sein de la simulation sont meilleurs en utilisant la technique de EM que celle des MMC. Toutefois, avec les MMC, bien que les comportements des agents manquent de réalisme d'un point de vue microscopique, ils exploitent beaucoup plus l'environnement que ceux issus de EM. Les résultats de la simulation au niveau macroscopique seront présentés dans le chapitre 4.

3.4 Discussion et conclusion

Dans ce chapitre, nous avons exposé une méthodologie de conception de comportements de déplacements d'agents pour une simulation destinée à reproduire des comportements réels observés. Deux techniques d'apprentissage ont été testées, pour apprendre le modèle le plus fidèlement possible, sur une base de données extraite à partir de séquences vidéo.

La première technique se base sur l'algorithme EM, capable d'estimer des comportements prototypes simples sous forme de trajectoires moyennes. A partir de ces déplacements types, il est possible de reproduire approximativement la majorité des trajectoires réelles simples, notamment de type "se déplacer de la porte d'entrée à la machine à café" ou "passer d'un couloir à l'autre". La capacité du modèle EM à regrouper correctement les trajectoires de déplacements a été vérifiée en utilisant la classification ascendante hiérarchique mesurant les proximités des trajectoires pour chaque cluster appris.

La deuxième technique se base sur les chaînes de Markov cachées. La structure du modèle MMC, composée d'états cachés et d'observations, permet de rassembler tous les comportements observés en un seul et unique modèle. Ainsi, cette structure générique apprise permet d'éviter l'inconvénient rencontré avec EM qui consiste à ne pas permettre aux agents de réaliser une succession de comportements. Effectivement, grâce à MMC, on est capable de combiner plusieurs comportements au sein d'un même agent. Il est à noter que les comportements de déplacement, transmis aux agents, sont traduits en motivations suivant un partitionnement relatif à des RI de l'environnement, connu *a priori*. De ce fait, chaque agent possède une liste d'objectifs à atteindre définissant leur parcours dans l'environnement simulé. Cependant, ces comportements peuvent devenir localement incohérents et sont fortement dépendants du nombre de régions d'intérêt à parcourir.

On constate aussi que le MMC, dans sa mise en pratique, est difficile à expérimenter dans la mesure où certains paramètres sont choisis empiriquement, contrairement à EM. Par conséquent, un passage à un problème plus compliqué, un espace plus grand ou des comportements plus complexes et plus hétérogènes rendent les MMC difficilement applicables. Notamment, dans un espace de vente, modéliser les comportements de déplacement des consommateurs avec un MMC produirait vraisemblablement des résultats médiocres.

Ainsi, la technique de EM semble plus adaptée à ce type d'environnement et de comportement. Elle permettra de discerner les différents prototypes de consommateurs pour reproduire de façon régulière leurs comportements de déplacement. Toutefois, l'un des objectifs de la simulation est aussi de retrouver les régions les plus fréquentées et les moins fréquentées dans l'espace de vente selon un agencement donné. Même si les déplacements des agents au niveau micro peuvent paraître aberrants et non crédibles, il peut être intéressant d'évaluer le comportement du modèle EM et du modèle MMC au niveau macro. Ainsi, nous étendons notre analyse des deux techniques jusqu'à la vérification et la validation du résultat final de la simulation.

Dans le chapitre suivant, nous allons reporter, analyser et discuter les résultats obtenus par la simulation multi-agents dont les comportements des agents ont été conçus par la méthode de EM, d'une part, et de MMC d'autre part. Le résultat de la simulation représentant une carte de chaleur sera évalué par comparaison avec la carte de chaleur des données collectives initiales. La mesure d'un coefficient de corrélation contribuera à évaluer si le résultat de la simulation est fidèle au comportement collectif réel et donc valider le modèle, ou dans le cas contraire, remettre en cause sa conception, les données utilisées ou les paramètres des techniques d'apprentissage utilisés.

Chapitre 4

Implémentation et résultats

Sommaire

4.1	Introduction	81
4.2	Implémentation	82
4.2.1	Plateformes de simulation multi-agents	82
4.2.2	Description du workflow lié à notre méthodologie	87
4.3	Résultats	92
4.3.1	Résultats avec EM	92
4.3.2	Résultats avec MMC	94
4.3.3	Synthèse	97
4.4	Transposition au commerce	98
4.5	Conclusion	102

4.1 Introduction

Dans le chapitre précédent, nous avons proposé une méthodologie de conception de simulation multi-agents se basant uniquement sur l'observation du monde réel. Nous nous sommes concentrés, particulièrement, sur la phase d'agentification. Cette phase consiste à extraire des comportements d'agents qui sont caractérisés par des trajectoires réelles observées dans des séquences vidéo. Nous avons procédé, en l'occurrence, à l'identification de typologies d'agents selon leurs comportements de déplacement. Leurs modèles de comportement ont été déterminés de façon quasi automatique selon deux techniques différentes, l'une utilisant EM, l'autre utilisant les MMC.

Dans le but de ne pas faire suivre aux agents des trajectoires stéréotypées et ainsi autoriser une flexibilité dans leurs déplacements, nous avons été amenés à traduire les trajectoires extraites des vidéos en listes de motivations. Ces listes définissent des successions de régions d'intérêt (de l'environnement) visitées. Par conséquent, dans le cas où un réaménagement de l'environnement est envisagé, les motivations suivront naturellement la nouvelle répartition des régions d'intérêt.

Dans ce chapitre, nous étudions de plus près la phase de simulation considérant son implémentation et ses résultats. Pour ce faire, nous reprenons et poursuivons l'étude sur le cadre applicatif

exploité dans le troisième chapitre. Pour rappel, l'environnement étudié représente le hall d'entrée d'un bâtiment universitaire équipé de caméras nous permettant d'enregistrer des trajectoires à partir desquelles seront extraites des listes de motivations. À partir de ces listes, différents comportements de déplacement des agents seront instanciés au sein de la simulation.

L'objectif final de la simulation est de construire incrémentalement une carte d'occupation de l'environnement (carte de points chauds / points froids) permettant de mettre en évidence les régions les plus convoitées de l'environnement observé, ainsi que les régions les plus inoccupées.

L'implémentation de notre proposition faite au chapitre 3 requiert une plate-forme de simulation multi-agents. Le choix de cette plateforme ainsi que son utilisation dans le cadre de notre problématique sont donc discutés dans une première partie de ce dernier chapitre. Dans une seconde partie, nous présentons les résultats de la simulation obtenus à la suite de différentes expérimentations mettant en jeu les deux techniques d'apprentissage que sont EM et les MMC. Nous terminons ce chapitre sur la manière d'utiliser notre méthodologie dans un cadre de simulation d'espace de vente.

4.2 Implémentation

Nous abordons, dans cette section, l'implémentation de l'approche présentée au chapitre 3. Nous commençons par un état de l'art des différentes plateformes utilisées pour les simulations multi-agents. Nous détaillons ensuite l'utilisation de la plateforme multi-agents retenue.

4.2.1 Plateformes de simulation multi-agents

Avec la démocratisation des simulations multi-agents, de nombreuses plateformes et outils de développement ont vu le jour. Parmi ces outils et plateformes les plus utilisés dans le monde académique, nous pouvons citer Swarm [Minar *et al.*, 1996], MASON [Luke *et al.*, 2005], NetLogo [Tisue et Wilensky, 2004], Breve [Klein, 2003] et Repast Symphony [North *et al.*, 2005]. Il est à noter que la plupart de ces outils est open-source et présente donc des intérêts en termes de coût, d'accessibilité au code source et d'intégration facile de nouvelles fonctionnalités. Sans se limiter au monde académique, nous exposons brièvement dans ce qui suit les avantages et inconvénients de plusieurs solutions et concluons sur notre choix.

4.2.1.1 Etat de l'art des outils existants

Cette section est une synthèse des logiciels de simulation multi-agents couramment rencontrés ; l'objectif étant de fournir une comparaison des plateformes pouvant être utilisées dans le cas de notre étude.

AnyLogic¹ est un outil de simulation commercial développé par la société XJ Technologies. Il supporte différents types de modélisation, notamment à base d'agents. Ce logiciel offre de nombreuses fonctionnalités permettant un développement rapide de simulations grâce à l'existence de bibliothèques d'objets permettant d'incorporer rapidement des éléments de simulation prêts à l'emploi. Cela permet le développement d'environnements de simulation visuellement riches et interactifs.

1. <http://www.anylogic.com/>

Cependant, à l'instar de la majorité des logiciels payants, les codes sources ne sont pas disponibles et il n'existe pas de support direct permettant aux utilisateurs de s'entraider aisément.

Swarm² [Minar *et al.*, 1996] est une plateforme logicielle multi-agents pour la simulation de systèmes adaptatifs complexes. Swarm fournit des bibliothèques orientées objet de composants réutilisables pour construire des modèles, analyser, afficher et contrôler des expériences sur ces modèles. Cependant, seulement quelques exemples de modèles sont disponibles et d'importants investissements sont nécessaires pour se familiariser avec la construction de modèles.

NetLogo³ [Tisue et Wilensky, 2004] est un langage de programmation multi-agents et un environnement de modélisation pour la simulation de phénomènes naturels et sociaux. Il est assez simple d'usage et adapté pour la modélisation de systèmes complexes dynamiques. Cependant, la contrepartie de cette simplicité d'usage est que des modèles d'agents complexes sont difficilement simulables.

La plateforme Breve⁴ [Klein, 2003] est un environnement conçu pour la simulation de systèmes multi-agents réalistes en 3D. Breve peut être utilisée comme un outil pour explorer tout type de monde simulé. Il facilite la construction de simulations à base d'agents complexes en gérant automatiquement la communication entre agents, la représentation dans l'espace 3D, le rendu graphique, la simulation physique, etc. Il se base cependant sur une langage exotique : le langage Steve⁵.

MASON⁶ [Luke *et al.*, 2005] est une bibliothèque en Java pour des simulations multi-agents à événements discrets, fournissant suffisamment de fonctionnalités pratiques répondant à de nombreux besoins de simulation. MASON peut être configuré pour garantir la portabilité, ce qui signifie que les paramètres de la simulation peuvent produire les mêmes résultats quel que soit le système d'exploitation. Cependant, la complexité d'implémentation de modèles sous cette plateforme n'est pas négligeable.

Repast Symphony⁷ [North *et al.*, 2005] acronyme de REcursive Porous Agent Simulation Toolkit est un outil de simulation développé en Java et intégré à Eclipse. Le terme "REcursive" fait allusion à la capacité de modéliser des combinaisons complexes imbriquées d'agents et d'espaces, et "Porous" fait référence à la capacité de modéliser des agents qui sont distincts, mais pas nécessairement atomiques. Il permet la modélisation multi-agents en simplifiant la création du modèle et son utilisation. Repast Symphony offre aux utilisateurs une grande variété de fonctionnalités comme par exemple le couplage avec des outils tels que Weka ou Matlab. Il existe, toutefois, différentes versions de la plateforme Repast (Repast3, Repast Symphony, Repast HPC,...) non compatibles les unes par rapport aux autres. A chaque nouvelle fonctionnalité, un nouvel outil est créé.

2. développé par Santa Fe Institute/SWARM Development Group, USA (http://www.swarm.org/index.php/Main_Page)

3. développé par Centre for Connected Learning and Computer-Based Modelling, Northwestern University, USA (<http://ccl.northwestern.edu/netlogo/>)

4. développée par Jon Klein (<http://www.spiderland.org/breve>)

5. un langage orienté objet, s'inspirant des langages C, Smalltalk et Objective C

6. développé par le laboratoire Evolutionary Computation de l'université George Mason et du centre GMU pour Social Complexity, USA (<http://cs.gmu.edu/eclab/projects/mason>)

7. développé par University of Chicago, Department of Social Science Research Computing, USA (<http://repast.sourceforge.net>)

Dans l'optique de se décider sur l'outil le plus adéquat à nos besoins, nous avons privilégié certaines propriétés présentées dans le tableau 4.1. Nous avons, par exemple, favorisé le langage de programmation Java au reste, en raison de sa simplicité et de son intelligibilité. Nous avons également privilégié un outil permettant de simuler indifféremment un environnement discret ou continu. Enfin, l'outil doit être fort de son support technique grâce à une communauté active et permettre certaines flexibilités telles que l'intégration d'outils externes ou de bibliothèques d'algorithmes utiles dans le domaine de l'intelligence artificielle en général.

Outils	Support environnement continu (c) et discret (d)	Java	Interopérabilité avec les outils de data mining (Weka, Matlab...)	Passage à l'échelle	Communauté active	Bibliothèque d'algorithmes évolutionnaires et d'apprentissage
Anylogic	d/c	oui	oui	oui	oui	oui
Swarm	d/c	oui	?	oui	oui	?
NetLogo	d	non	?	non	oui	?
Breve	d/c	non	oui	?	non	oui
Mason	d/c	oui	non	?	oui	non
Repast Symphony	d/c	oui	oui	oui	oui	oui

TABLE 4.1 – Comparaison des outils de simulation multi-agents.

A l'issue de notre état de l'art, notre choix s'est porté sur Repast Symphony [Railsback *et al.*, 2006]. Plusieurs raisons justifient ce choix. D'une part, sa licence laisse libre accès à son code, d'où la transparence de son fonctionnement interne. D'autre part, l'utilisation du langage Java nous offre une certaine facilité de programmation. De plus, Repast Symphony fournit un environnement de développement permettant la modélisation d'un ensemble d'agents, dotés d'un certain nombre de comportements et de propriétés dont les interactions.

4.2.1.2 Utilisation de Repast Symphony

La structure standard de modélisation dans Repast Symphony est basée sur deux concepts essentiels [Howe *et al.*, 2006] :

- le contexte : qui est une sorte de conteneur englobant tous les composants du modèle, y compris les agents. Il ne fournit, cependant, aucune information directe sur l'espace ou le mécanisme d'interaction entre les agents. Il décrit, de ce fait, une méta-population dans une infrastructure basique et fournit l'état relatif aux conditions des agents prototypes qui existent dans le contexte ;
- les projections : qui peuvent créer une structure rattachée à un contexte, permettant ainsi de définir les relations entre les agents qui appartiennent à ce contexte. Ils définissent, en général, la nature de l'espace dans lequel les agents opèrent. Il existe trois types de projections (voir figure 4.1) :
 - projection GIS : où l'agent possède une position et peut se déplacer dans un environnement géographique (environnement continu) ;

- projection Grid : où l'agent évolue dans un environnement discret de type grille ;
- projection réseau : spécifie les relations entre agents, notamment dans les réseaux sociaux.

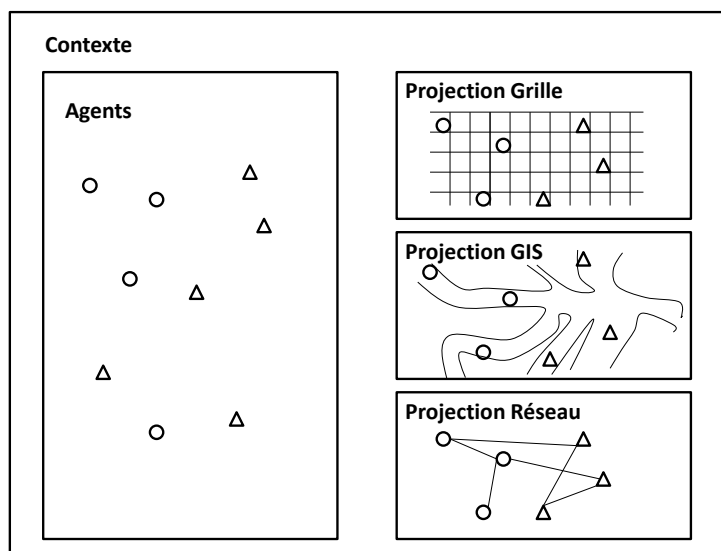


FIGURE 4.1 – Modèle de développement sur Repast Symphony

Nous avons choisi, pour notre simulation, la projection de type grille pour la simple raison que son utilisation est la moins complexe, sachant que le rendu visuel, au moment de l'exécution de la simulation, ne représente pas une priorité dans notre travail.

4.2.1.3 Architecture de la simulation

La structure de la simulation est principalement répartie sur trois éléments définissant :

- une classe pour le contexte général : "EnvironmentContext", qui s'occupe de la gestion du lancement de la simulation ;
- une classe définissant les agents appelés "HumanAgent" ;
- une classe représentant le gestionnaire de terrain que nous nommons "Mediator". Cette classe est créée pour maintenir une certaine genericité en dissociant les déplacements de la nature de l'environnement (discret ou continu), d'où l'appellation de médiateur.

Par ailleurs, les agents ont accès à leurs motivations disponibles soit à partir des résultats du Clustering (EM), soit à partir du MMC appris. La structure de la simulation est présentée dans la figure 4.2.

La génération d'un agent dans la simulation se fait au niveau du gestionnaire de la simulation une fois le type d'environnement défini (discret ou continu). Le nombre d'agents impliqués dans la simulation est égal au nombre initial d'individus observés dans l'environnement réel. Dans le cas où les comportements des agents (leurs motivations) sont obtenus par clustering, la répartition des agents est définie selon le pourcentage de chaque classe obtenu. Dans le cas où les comportements

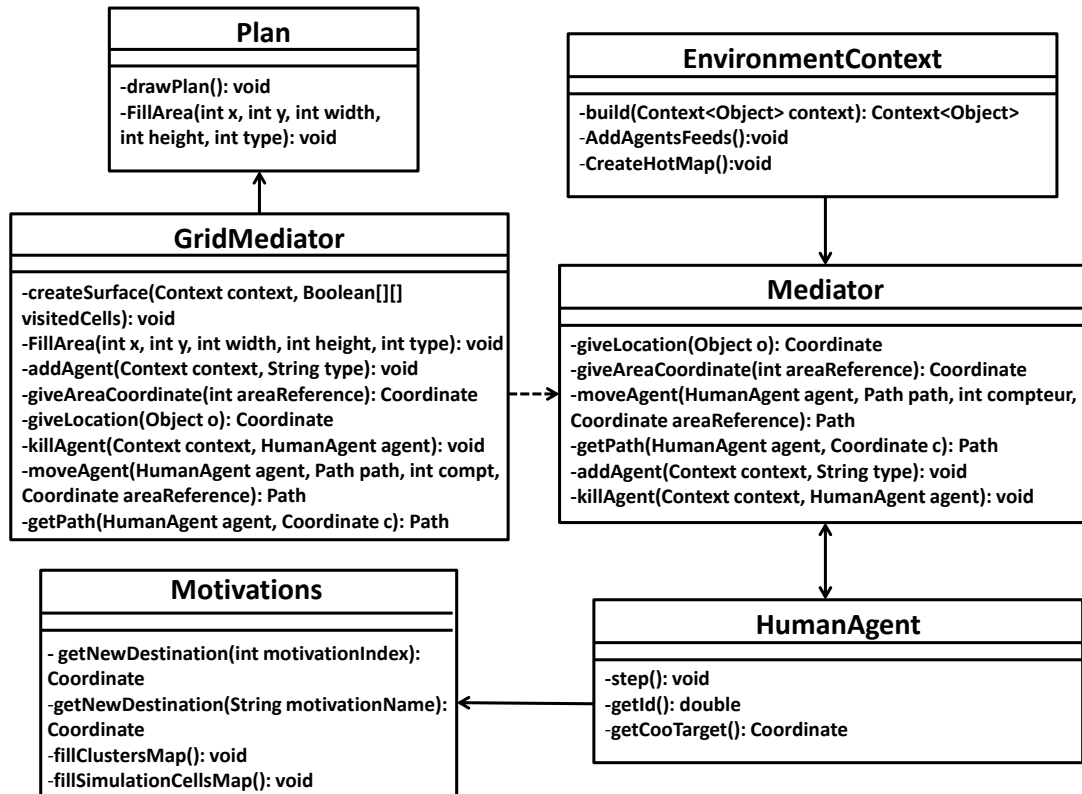


FIGURE 4.2 – Structure générale du modèle de la simulation

des agents sont obtenus par apprentissage d'un MMC, chaque agent possède ses propres objectifs générés à partir de la matrice d'observations, la matrice de transitions et le vecteur initial du MMC appris.

Le gestionnaire de terrain ("Mediator") est en charge du déplacement des agents. Il récupère, à chaque pas de temps, l'objectif courant de l'agent et le traduit en coordonnées de grille (choisies aléatoirement) appartenant à la région correspondante à l'objectif. Grâce à cette position, le gestionnaire déplace l'agent de sa position initiale à sa destination en utilisant un algorithme de plus court chemin. Par ailleurs, si l'algorithme EM est appliqué, un temps consacré à chaque motivation (région d'intérêt dans l'environnement) est pris en compte. L'agent reste immobile à chaque destination pendant t pas de temps, tel que t représente le temps de chaque motivation (pour rappel voir tableau 3.2). L'avantage de prendre en compte le temps, dans ce cas de figure, est d'accentuer l'intérêt de certaines régions par rapport à d'autres. En effet, un agent qui marque une longue pause à un certain endroit peut manifester des indications sur son objectif. Par exemple, si un agent s'arrête un moment à côté de la machine à café, il est très probable que cet agent attende la préparation de son café.

Enfin, le gestionnaire se charge également de créer au fur et à mesure de la simulation, une carte des points chauds dépendants des déplacements des agents. Cette carte nous permet de visualiser

les zones les plus fréquentées durant la simulation.

4.2.2 Description du workflow lié à notre méthodologie

Dans ce qui suit, nous exposons les différentes étapes de notre méthodologie mettant en avant les fichiers d'entrée et de sortie à chaque phase de traitement. Nous procédons, toutefois, selon deux manières différentes par rapport à l'obtention des données relatives à l'environnement. La différence réside dans le découpage de l'environnement. D'un côté, il s'agit d'appliquer la méthode de découpage "manuelle" telle que présentée au troisième chapitre au niveau de la section 3.2.2.1. D'un autre côté, il est question d'une autre méthode permettant un découpage de l'environnement d'étude, de façon automatisée, reposant sur la méthode de Voronoï. Cette technique est présentée en annexe A et traite uniquement de l'environnement vue caméra, ce qui contraint à réaliser une simulation sur un environnement distordu et difficile à manier. L'idée d'appliquer cette technique est de rendre la méthodologie entièrement automatisée y compris les données de l'environnement, qui se trouvent fortement liées à l'application. Bien que le découpage soit différent, il permet néanmoins d'automatiser cette tâche à une certaine limite. De ce fait, nous utilisons le découpage "manuel" comme méthode principale, et le découpage suivant "Voronoi" comme méthode alternative venant appuyer les résultats obtenus.

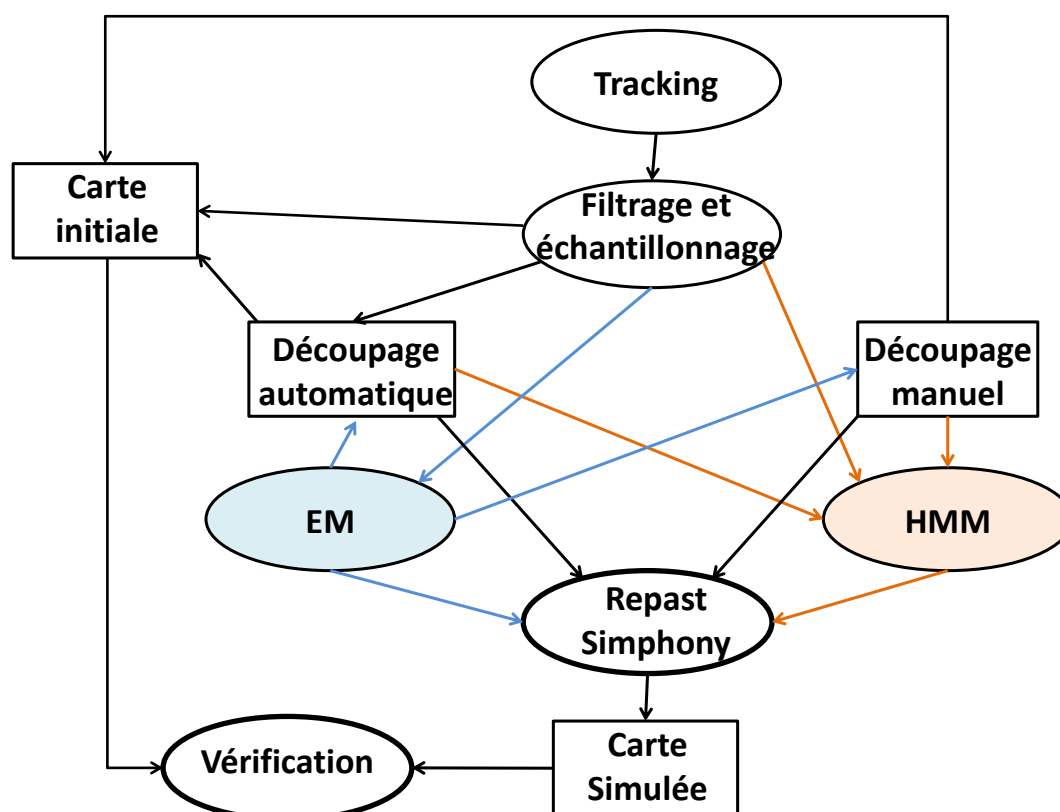


FIGURE 4.3 – Workflow de la méthodologie retenue

4.2.2.1 Vue globale

La figure 4.3 montre une synthèse des traitements réalisés depuis l'acquisition des séquences vidéo jusqu'à la vérification des résultats de la simulation. Nous décrivons, dans ce qui suit, chacune des opérations réalisées.

Le module de tracking

Ce module a pour objectif l'extraction des trajectoires des personnes à partir du flux vidéo. En effet, partant d'un ensemble de séquences d'images provenant d'une caméra filmant pendant une journée entière, nous obtenons, suite à un traitement approprié, des informations associées aux trajectoires extraites des personnes ayant parcouru la scène. Pour ce faire, à partir du flux vidéo, l'algorithme extrait les formes mobiles de l'image. Cette première opération est réalisée sur la base d'une extraction de fond (le fond étant alors assimilé à la partie statique de l'environnement enregistrée dans une configuration de référence). Aux formes extraites, modélisées sur la forme de blobs (agrégats de pixels sans forme définie *a priori*, voir figure 4.4), sont associés des vecteurs de caractéristiques.

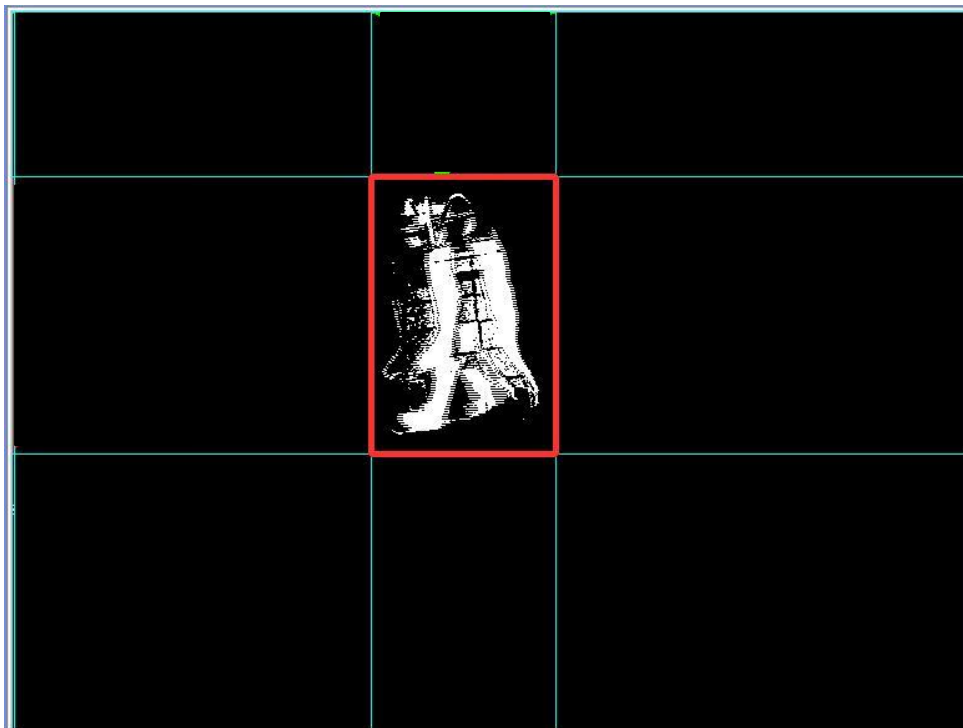


FIGURE 4.4 – Illustration d'un blob dans l'environnement étudié

A l'aide d'une métrique adaptée, les objets d'intérêts sont mis en correspondance trame après trame, permettant ainsi de reconstituer les trajectoires recherchées. Une trajectoire est donc modélisée par la suite des positions occupées au cours du temps. Le temps est lui-même repéré par l'indice de la trame (c'est-à-dire de l'image) acquise par la caméra, l'origine correspondant à l'instant de démarrage de la prise de vue. Les positions prises en considération correspondent à la projection des

centres de gravité des entités d'intérêt (*a priori* des personnes) dans l'image. Nous reportons, de ce fait, dans des fichiers :

- le nombre de trajectoires extraites ;
- la suite des positions occupées pour chaque trajectoire, chacune associée à un numéro de trame ;
- un identifiant de forme détectée ;
- une suite de vecteurs de caractéristiques telles que la boîte englobante du blob, le vecteur de direction, etc.

L'objectif principal des traitements associés à ce module étant la constitution d'un ensemble de trajectoires "individualisées", nous ne nous intéressons qu'aux coordonnées des déplacements individuels.

Le module de filtrage et échantillonnage des données

Le premier rôle de ce module est de réaliser le filtrage de l'ensemble des trajectoires obtenues dans le module de tracking. Il existe, en effet, des défauts au niveau du tracking tels que des phénomènes d'occultation de personnes ou la détection de mouvements erronés, qui mènent à des trajectoires aberrantes, généralement trop courtes. Ainsi, il est nécessaire d'éliminer ce type de trajectoires non utilisables. Par conséquent, les trajectoires comportant moins de 30 points sont éliminées. Par ailleurs, suite à ce filtrage des trajectoires, le module permet également de réaliser un échantillonnage afin de normaliser la longueur des trajectoires à 50 points. Ce choix correspond à une moyenne sur l'ensemble des trajectoires enregistrées. Le reste de notre étude se base sur l'ensemble de ces trajectoires filtrées et normalisées (soit au total 682 sur notre cas d'étude). À partir de cette étape, nous distinguons deux traitements possibles permettant l'apprentissage de motivations d'agents : une première est basée sur EM, la seconde sur les chaînes de Markov cachées.

Le module de clustering avec EM

Les 682 instances de trajectoires, filtrées et échantillonnées à 100 attributs (50 coordonnées), représentent la base d'apprentissage pour l'algorithme EM (détaillé dans la section 3.3.1). Les résultats de la classification, obtenus via l'utilisation du logiciel Weka [Witten *et al.*, 1999], donnent un ensemble de clusters (24 dans le cadre de notre exemple). Chaque cluster est caractérisé par :

- le nombre (et la proportion) de trajectoires dans chaque cluster ;
- la trajectoire moyenne ;
- la covariance des trajectoires associées à ce cluster.

Les trajectoires moyennes, ainsi obtenues, sont traduites en listes de région d'intérêts (RI) selon deux mécanismes différents.

- Découpage manuel :

La première façon de faire est d'utiliser un découpage "manuel" de l'environnement (voir la section 3.2.2.1). Les trajectoires sont alors converties, selon les coordonnées, en listes de RI identifiées et reportées dans un tableau à 5 colonnes : abscisse du coin supérieur de la zone, ordonnée du coin supérieur de la zone, largeur de la zone, hauteur de zone, label de la zone.

- Découpage automatique :

La seconde façon de faire est d'utiliser le découpage "automatique" reposant sur une adaptation des diagrammes Voronoï détaillée dans l'annexe A. Les RI sont identifiées, dans ce cas, par des

numéros distincts dans une cartographie où chaque région est représentée par un ensemble de numéros identiques.

Indifféremment de la façon dont les RI sont caractérisées, ensuite, chaque point de la trajectoire est labellisé par la région d'intérêt correspondante à la localisation du point. Une trajectoire, correspondante à une succession de points, est ainsi transformée en une liste de régions d'intérêt. Dans le cas où plusieurs labels successifs sont identiques, ceux-ci sont regroupés en un seul et leurs durées sont sommées afin de caractériser le temps passé dans la zone.

Les types d'agents à simuler sont reportés dans un tableau tel que celui présenté au chapitre 3 (tableau 3.2). Chaque ligne y représente un cluster définissant un prototype de comportement de déplacement d'agent. Elle est composée d'un pourcentage indiquant la représentativité de ce type d'agents par rapport aux autres et est suivie d'une liste de couples (régions d'intérêt, temps).

Le module d'apprentissage avec MMC

Toutes les instances des trajectoires filtrées et échantillonnées sont directement traduites en listes de RI, selon le même principe présenté dans le paragraphe précédent pour les trajectoires moyennes. Toutefois, la notion de temps, telle qu'elle est considérée avec EM, ne peut être ici prise en compte. En effet, considérer des couples (RI, temps) aurait induit à un grand nombre d'observations sans facteur de similarité, rendant de ce fait difficile l'apprentissage du MMC. La liste des séquences d'observations représente, de ce fait, la base d'apprentissage des comportements de déplacement pour le MMC. Nous obtenons, au final, les paramètres du modèle MMC appris, notamment la matrice de transitions A , la matrice d'observations B et le vecteur π . La détermination des listes de RI des agents découle de ces paramètres suivant l'algorithme 6 de génération de séquence d'observations.

Algorithm 6 Algorithme de génération de séquence d'observations à partir d'un MMC

- 1- Choisir l'état initial i selon la distribution initiale des états π .
 - 2- Fixer $t = 1, T$.
 - 3- Choisir O selon $b_i(k)$, la distribution de probabilité d'observations à l'état i .
 - 4- Choisir j selon $\{a_{ij}\}$, $j = 1, 2, \dots, N$ la distribution de probabilité de transition depuis l'état i .
 - 5- Fixer $t = t + 1$, retourner à l'étape 3 si $t < T$, sinon finir la procédure.
-

Le module de simulation

Selon le mécanisme d'apprentissage choisi (EM ou MMC), les motivations des agents, construites depuis les séquences de RI provenant d'EM ou à partir des paramètres du MMC appris, sont incorporées dans le traitement de la simulation. D'autres paramètres sont également pris en compte, notamment l'emplacement des régions et les labels correspondants aux RI sur le plan simulation (correspondant à un découpage manuel ou suivant Voronoï).

Au niveau de l'interface du simulateur de Repast Symphony, comme le montre la figure 4.5, nous disposons de plusieurs onglets. Partant de gauche à droite, nous retrouvons :

- une fenêtre de l'arbre du scénario où on charge la classe contexte de la simulation ainsi que les types d'agents, leurs représentations, le type de l'environnement et sa représentation ;
- une fenêtre de paramètres permettant d'agir sur certaines variables de la simulation telles que le nombre d'agents intervenants ;

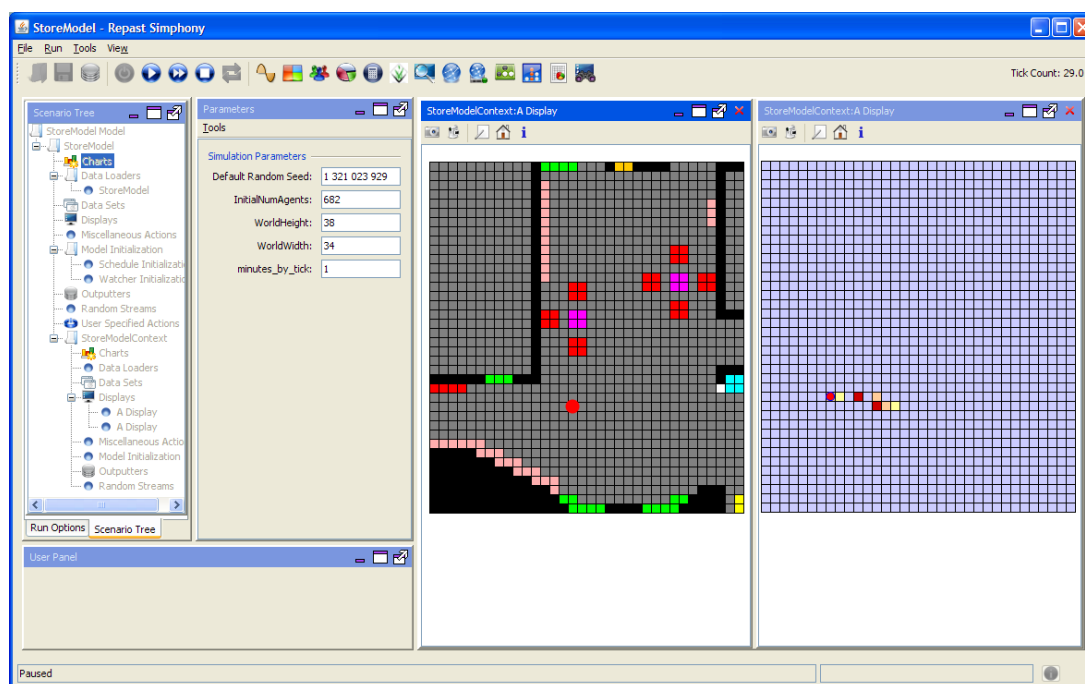


FIGURE 4.5 – L’interface de Repast Simphony pour la simulation du hall d’entrée du bâtiment

- une fenêtre de visualisation de la simulation ;
- une fenêtre de visualisation de la création de la carte des points chauds et des points froids au fur et à mesure de l’exécution de la simulation.

Ce dernier point représente le résultat de la simulation quand son exécution s’achève. La carte obtenue est récupérée afin de vérifier son niveau de fidélité par rapport à la réalité.

Le module de vérification

Dans le but d’évaluer la qualité du résultat de la simulation, nous comparons la carte des points chauds et des points froids obtenus en simulation à la carte “réelle” (provenant des images de la caméra et représentée par la carte initiale sur la figure 4.3). La similitude entre ces deux cartes est évaluée par un calcul de coefficient de corrélation. L’obtention d’un coefficient inférieur à un certain seuil ainsi qu’une interprétation visuelle de la carte résultante, établie par un expert, permettent de valider ou de remettre en cause la simulation.

Nous venons d’exposer le workflow complet de notre méthodologie. Nous avons également traité les détails de chaque module depuis le traitement d’images jusqu’à la simulation. Dans la section qui suit, nous détaillons les résultats obtenus.

4.3 Résultats

À titre de rappel, le but de la simulation, telle qu'elle a été conçue, est de parvenir à générer une carte de chaleur (la carte des points chauds et des points froids) retraçant les zones les plus fréquentées et les moins fréquentées dans l'environnement étudié. L'information prodiguée par cette carte permet de mettre en évidence la qualité d'aménagement ou d'agencement de l'espace selon l'importance des régions exploitées ou très peu visitées. Pour être en capacité d'exploiter correctement la carte de chaleur obtenue, il faut s'assurer de sa qualité. Nous évaluons donc, dans cette section, le degré de réalisme des cartes de chaleur, obtenues suite à l'emploi d'EM et des MMC, à la carte de chaleur "réelle" provenant de l'ensemble des trajectoires initiales avant traitement, et ce, au moyen d'un calcul de coefficient de corrélation détaillé en section 3.2.3 du chapitre 3. Nous exposons, dans ce qui suit, les résultats de la simulation et leur évaluation, liés à EM et aux MMC respectivement.

4.3.1 Résultats avec EM

Dans un premier temps, nous avons appliqué notre méthodologie mettant en œuvre la technique de EM pour la classification des trajectoires. Les déplacements modèles ainsi obtenus sont, ensuite, parcourus par les agents après avoir été traduits en liste de motivations. Afin d'obtenir une carte de chaleur simulée comparable à la carte "réelle", nous avons généré 682 agents (qui est le nombre d'instances de trajectoires extraites à partir desquelles la carte "réelle" a été générée). La répartition des différents types d'agents impliqués dans la simulation est proportionnelle à la distribution des classes, obtenues avec EM (cf 3.2).

Au cours de la simulation, à chaque pas de temps un agent est généré pour accomplir son comportement de déplacement. Une fois la simulation exécutée, la carte de chaleur résultante peut être évaluée. Pour ce faire, nous examinons la pertinence de cette carte visuellement, selon les connaissances que nous possédons relativement à l'environnement ainsi que par comparaison à la carte "réelle" [Saffar *et al.*, 2012]. Ensuite, nous procédons au calcul de coefficient de corrélation entre les deux cartes.

Sur la figure 4.6, nous constatons que les zones B, C, E, F sont similaires dans les deux cartes (réalité et simulation), et correspondent aux RI principales de l'environnement. En effet, nous retrouvons l'espace café (E), la sortie des classes (F) et l'entrée principale (B) comme étant les régions les plus parcourues dans la simulation. Par ailleurs, la carte 4.6(a) vient corroborer cette observation, démontrant des concentrations au niveau de ces mêmes régions. De même pour la région centrale (C), qui représente une zone de croisement entre le flux orienté nord-sud et celui orienté est-ouest, se retrouve dans les deux cartes.

En revanche, les zones A et D diffèrent. Cela peut s'expliquer par plusieurs constatations. D'une part, à l'issue du clustering, bon nombre des trajectoires passant par les zones A et D se sont retrouvées regroupées avec des trajectoires, certes proches, mais plutôt situées dans l'espace entre les zones B et C. D'autre part, du point de vue de la simulation, les algorithmes utilisés pour les évitements d'obstacles ont également contribué à éloigner les agents des zones A et D très obstruées par des chaises. Ces différences peuvent s'interpréter comme une simplification du monde réel (qui est un des buts de la simulation) plutôt que comme un défaut de modélisation. Notons qu'il aurait

été plus gênant de faire apparaître en simulation des points chauds qui n'existent pas en réalité, chose que nous n'avons pas constatée à l'issue de nos expérimentations.

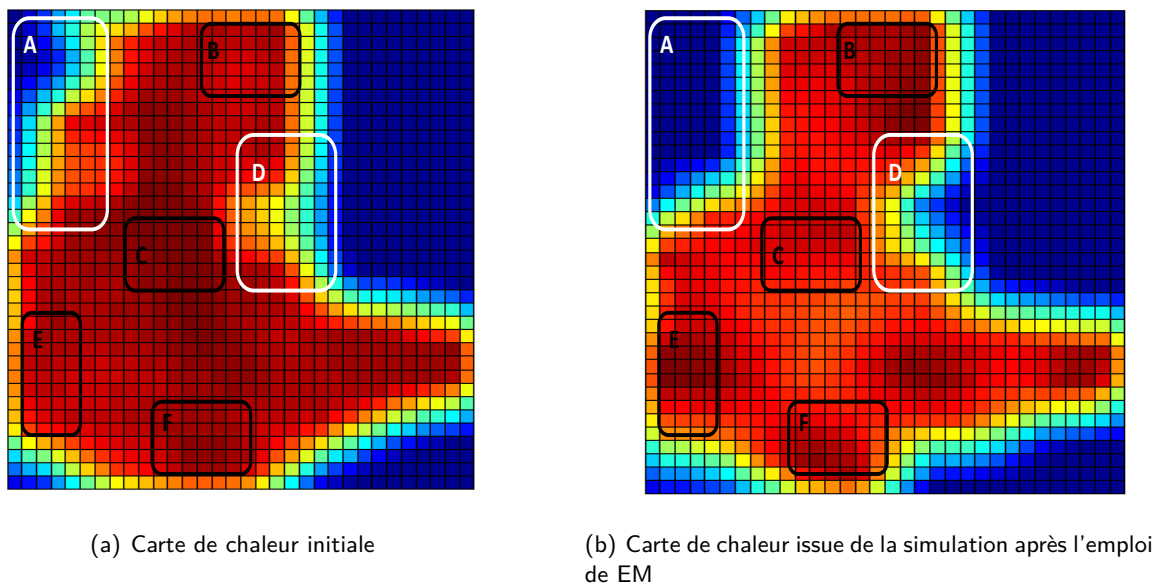


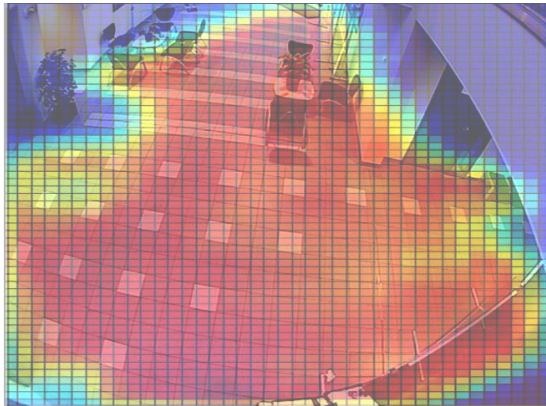
FIGURE 4.6 – Comparaison entre la carte de chaleur initiale et celle après l'emploi de EM

Ayant visuellement un résultat plutôt favorable, nous souhaitons corroborer cette observation par une métrique utilisée, généralement, dans la reconnaissance d'image, représentant le coefficient de corrélation. Cette métrique mesure une relation linéaire entre les intensités de deux images.

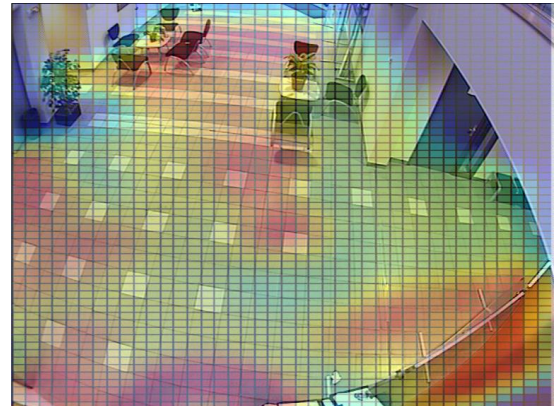
Notons CR la carte de chaleur réelle et CS la carte de chaleur simulée. Nous obtenons un coefficient de corrélation $r_{CR,CS} = 0.9327$. Ce coefficient est très élevé. Ce qui confirme le fait que les deux cartes soient très semblables.

Pareillement, le résultat de l'évaluation de la carte de chaleur simulée sur le plan, vue de dessus, de la simulation, est satisfaisant. Néanmoins, quand il s'agit de simulation en vue caméra, les différences sont plus accentuées. La figure 4.7 représente une comparaison entre la carte initiale vue caméra et la carte simulée, également vue caméra. Nous les avons superposées sur l'environnement réel afin de mieux visualiser les régions les plus convoitées.

Nous remarquons que les zones représentant des RI principales sont bien mises en évidence, particulièrement sur la carte simulée dans la figure 4.7(b). Par exemple, les régions correspondant à la machine à café, l'entrée des classes et la sortie principale sont remarquablement rouges. D'autre part, nous notons un coefficient de corrélation $r_{CR,CS_{cam}} = 0.8479$ avec CR : carte de chaleur initiale sur le plan caméra, et CS_{cam} : carte de chaleur simulée sur le plan caméra, avec EM.



(a) Carte de chaleur initiale



(b) Carte de chaleur issue de la simulation après l'emploi de EM

FIGURE 4.7 – Comparaison entre la carte de chaleur initiale et celle après l'emploi de EM (Vue caméra)

Bien que le coefficient de corrélation, entre la carte réelle et la carte simulée, soit plus faible pour une simulation en vue caméra qu'une simulation en vue de dessus, le résultat reste convenable. Il permet de révéler, dans les deux cas, des indications instructives sur l'environnement et ses différentes régions d'intérêt principales.

Nous pouvons conclure, par conséquent, que l'application de la technique d'EM dans la classification des trajectoires extraites, pour la conception d'une simulation multi-agents, permet de réaliser une simplification du monde réel tout en faisant apparaître clairement les zones les plus remarquables en terme de visites.

De la même manière, nous avons appliqué et testé la technique du MMC permettant de générer, dans la simulation, une population d'agents reproduisant des comportements individuels réels appris. La section suivante présente les résultats obtenus.

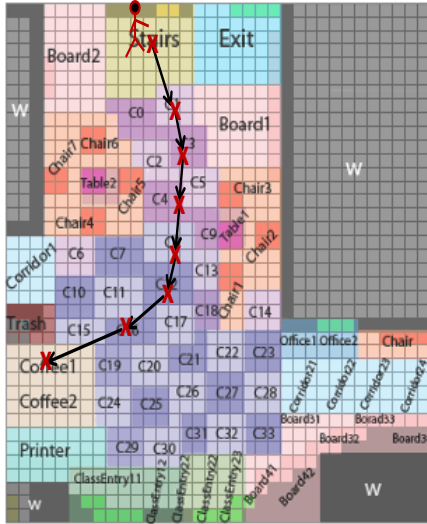
4.3.2 Résultats avec MMC

Selon l'échantillon d'instances de données d'apprentissage dont nous disposons, nous avons montré, dans la section 3.3.2, que le nombre optimal d'états cachés pour le MMC est de 20 états.

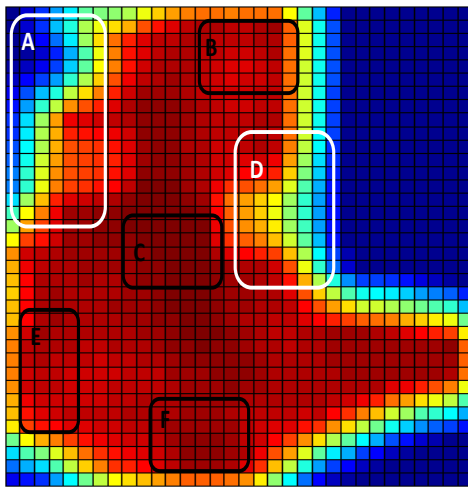
Chaque agent, parmi les 682 agents créés tout au long de la simulation, adopte un comportement de déplacement qui lui est propre. Ce comportement est particulier, car il provient de la structure probabiliste du MMC. Il est donc possible de générer d'innombrables listes d'observations différentes, conduisant à des comportements de déplacements tous différents.

Ces listes, représentant les motivations successives des agents, comportent 8 observations (voir explication au chapitre 3, section 3.3.2.2). À partir de ces dernières, on est capable de produire des trajectoires très proches des trajectoires réelles à partir desquelles le MMC a été appris. La figure 4.8(a) représente une telle trajectoire dont la séquence d'observations générée est : Stairs, C1, C3, C4, C8, C12, C16, Coffee1. De plus, d'autres formes de comportements de déplacement ont été

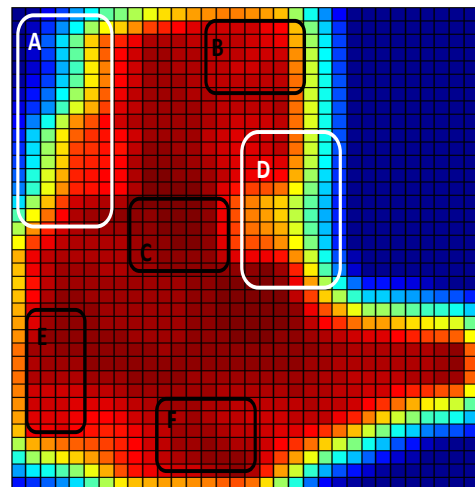
observées combinant au moins deux trajectoires réelles qui s'enchainent comme le montre la figure 4.8(b) et ayant la suite d'observations suivante : Stairs, C2, Chair5, C11, Coffee1, C20, C22, Office2. On constate également des comportements de déplacement qui manquent de cohérence. Notamment, il existe des cas où les agents franchissent plusieurs fois les mêmes régions ou réalisent des aller-retour entre deux régions ou plus. Ce qui ne correspond pas forcément à une attitude naturelle, observée sur les vidéos, comme on peut le voir dans la figure 4.8(c) pour le déplacement : Stairs, C0, C1, C4, Exit, C3, C2, Chair5.



Malgré ces comportements de déplacement non réalistes d'un point de vue micro, la carte de chaleur (point de vue macro) qui en découle reste convenable. La figure 4.9 affiche, côte à côte, la carte réelle (identique à la figure 4.6(a)) et, à droite, la carte simulée en ayant recours au MMC à 20 états cachés. Ces deux cartes sont sensiblement identiques. Nous avons gardé les mêmes zones de comparaison que nous avons préalablement utilisées dans la figure 4.6 pour l'étude de l'application d'EM. Nous observons que toutes les zones A, B, C, D, E et F dans les deux cartes sont similaires. Il existe, cependant, un passage entre la zone E et la zone F, en l'occurrence entre la machine à café et la sortie des salles de cours, légèrement plus dense dans la carte simulée. Ceci dénote que les agents possèdent une certaine abstraction par rapport à la réalité.



(a) Carte de chaleur initiale



(b) Carte de chaleur issue de la simulation après l'emploi de MMC à 20 états

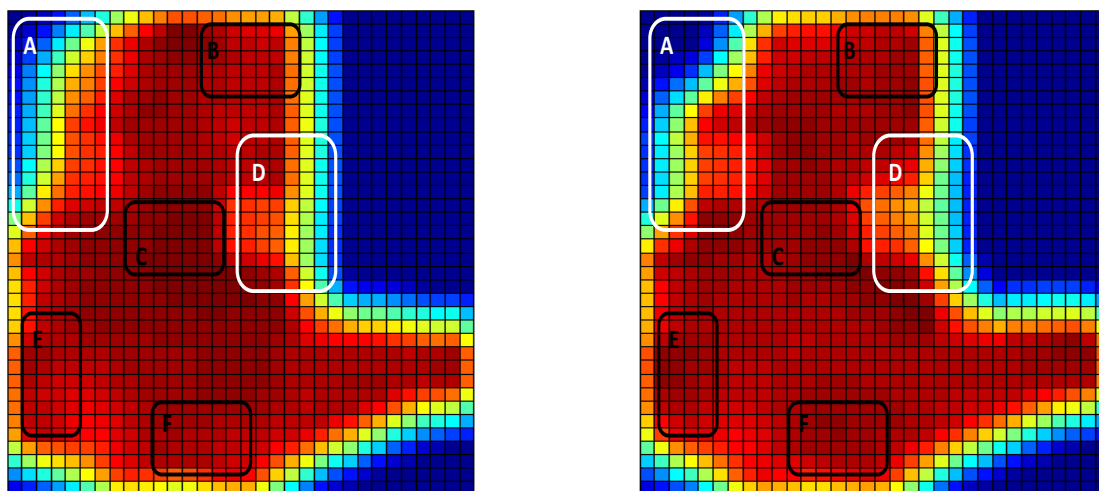
FIGURE 4.9 – Comparaison avec carte de chaleur du MMC retenu

Au vu de la ressemblance entre la carte de chaleur réelle CR et la carte de chaleur simulée suite à l'application de MMC à 20 états cachés CS_{MMC20} , nous calculons et obtenons un coefficient de corrélation de $r_{CR,CS_{MMC20}} = 0.9950$.

Afin de corroborer l'évaluation *a priori* du nombre d'états cachés présentée au chapitre 3 (section 3.3.2), nous avons effectué des simulations avec des MMC comportant un nombre d'états cachés différent de 20. Nous avons ainsi considéré dans nos simulations des MMC à 3 et 8 états cachés (correspondant au graphe illustré dans la figure 3.17 du chapitre 3) et calculé un coefficient de corrélation pour les cartes de chaleurs correspondantes.

La figure 4.10 affichant les deux cartes résultantes des MMC à 3 et 8 états cachés, montre que plus on augmente le nombre d'états cachés, plus on se rapproche du résultat désiré, i.e. la carte réelle. Néanmoins, ils se rapprochent considérablement du résultat idéal (celui du MMC à 20 états cachés). Leurs coefficients de corrélation sont les suivants : $r_{CR,CS_{MMC3}} = 0.9863$ et $r_{CR,CS_{MMC8}} = 0.9900$.

Ces résultats confirment l'évaluation *a priori* du nombre d'états cachés fixé à 20 au chapitre 3, ainsi que la méthode utilisée se basant sur le calcul des diamètres du premier ordre réduisant la dispersion des états.



(a) Carte de chaleur issue de la simulation après l'emploi de MMC à 3 états

(b) Carte de chaleur issue de la simulation après l'emploi de MMC à 8 états

FIGURE 4.10 – Cartes de chaleur des MMCs pour différents nombres d'états cachés

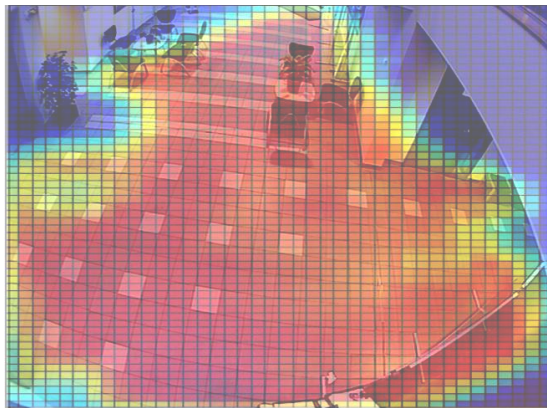
Après avoir comparé des cartes de chaleurs dans le plan simulation, nous allons faire les mêmes comparaisons dans le plan caméra. La figure 4.11 montre les deux cartes de chaleur sur le plan caméra dont celle à gauche représente la carte réelle et celle à droite représente la carte issue de la simulation pour un MMC à 20 états. Nous remarquons que les deux cartes sont moins similaires en comparaison des cartes sur le plan simulation. Nous retrouvons, néanmoins, les zones les plus chaudes (les plus visitées) notamment au niveau du centre, la sortie des classes, l'entrée principale ainsi que la zone de la machine à café. Le coefficient de corrélation de ces deux cartes est de $r_{CR,CS_{MMC20cam}} = 0.9438$. Aussi, à comparer avec les résultats de EM, la similarité entre ces deux cartes est plus importante.

4.3.3 Synthèse

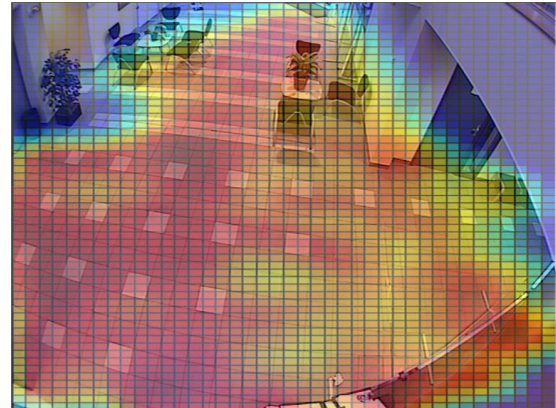
Ce que nous retenons des résultats obtenus est que les MMC, bien qu'ils ne permettent pas l'obtention d'un visuel cohérent (niveau micro) au moment de l'exécution de la simulation, aboutissent à un résultat final conforme à la réalité d'un point de vue macro.

Avec EM, le visuel des comportements de déplacement des agents au niveau local est correct. La carte de chaleur résultante, bien qu'elle semble un peu distincte de la carte réelle, fait clairement apparaître les informations essentielles à l'évaluation de la disposition de l'environnement.

D'autre part, quelle que soit la vue de la simulation, à savoir celle du dessus ou celle de la caméra, les scores obtenus par comparaison aux cartes du monde réel en termes de coefficient de



(a) Carte de chaleur initiale



(b) Carte de chaleur issue de la simulation après l'emploi de MMC à 20 états cachés

FIGURE 4.11 – Comparaison entre la carte de chaleur initiale et celle après l'emploi du MMC à 20 états cachés (Vue caméra)

corrélations sont quasi-similaires. Ce qui signifie que le découpage automatique, présenté en annexe, est solide. Il reste, néanmoins, effectif uniquement au niveau de la vue caméra. Or afin d'avoir une visibilité uniforme sur l'ensemble de l'environnement sans effets de perspective ni de distorsion et avoir des comportements interprétables, il est nécessaire de concevoir une simulation vue de dessus. Les déplacements des agents sont alors plus facilement intelligibles et interprétables.

Toutefois, le point fort des cartes réalisées sur le plan caméra est qu'elles ont été conçues à partir d'un environnement découpé de façon optimale, automatiquement, sans passer par une phase fastidieuse pour trouver la meilleure répartition des zones. Il aurait été intéressant de tester ce découpage sur une vue dans le plan du sol. Hélas, compte tenu de l'inclinaison de la caméra et de la distorsion de l'image, le calcul de l'homographie⁸ permettant de passer en vue de dessus n'était pas possible.

Nous sommes donc parvenus, sur un exemple simple, à modéliser et reproduire, au moyen d'une simulation et à partir de méthodes d'apprentissage, des comportements situés d'individus. Nous pouvons donc naturellement nous interroger sur la transposition de notre méthodologie à notre contexte applicatif initial : la simulation d'un espace de vente. La section suivante met en évidence les différentes étapes à mettre en œuvre pour utiliser notre méthodologie au cas plus complexe de la simulation d'un magasin.

4.4 Transposition au commerce

Nous avons, jusqu'à présent, appliqué et testé notre méthodologie sur un environnement simple. Or, le cadre applicatif que nous ciblons, la simulation d'un espace de vente, représente un environne-

8. i.e. une projection permettant de passer d'un plan à un autre.

ment plus complexe. Les RI principales sont potentiellement plus nombreuses et les comportements de déplacement des clients peuvent être plus variés et complexes.

Il serait donc plus approprié de focaliser la modélisation sur les principaux comportements de déplacement. Ce qui nous mène à privilégier l'utilisation de la technique de EM à celle du MMC. Nous présentons dans ce qui suit les étapes à suivre permettant de transposer notre méthodologie sur un espace de vente.

Supposons que nous ayons accès à partir d'un réseau de caméras à des séquences vidéo d'un magasin de vente. Afin d'obtenir la carte des points chauds / points froids du magasin, il est essentiel que le réseau de caméras couvre la totalité de l'espace de vente.

La première étape consistera à réaliser un tracking de tous les clients sur une période de temps fixée (un jour, une semaine, un mois,...) à partir des séquences vidéo. Cette étape nécessitera en particulier de garantir le suivi d'une même personne au travers du réseau de caméras. À la suite du tracking, nous disposerons des trajectoires brutes représentant les comportements de déambulation des clients observés. Ces trajectoires seront ensuite filtrées et rééchantillonnées de manière à construire la base de données d'apprentissage.

Cette base de données sera utilisée ensuite pour le prototypage des comportements de clients sur la base des techniques d'apprentissage présentées au chapitre 3. Parallèlement, disposant du plan du magasin ainsi que de l'organisation des produits, un partitionnement de l'environnement en régions d'intérêt pourra être réalisé. Chaque région se rapportera au type des produits environnants et sera annotée en conséquence avec l'aide possible d'un expert.

Suivant ce découpage, les trajectoires des "prototypes clients" seront traduites en listes de régions d'intérêt. Par la suite, une simulation multi-agents sera créée avec :

- un environnement s'appuyant sur le partitionnement annoté du plan du magasin ;
- les motivations des agents provenant des listes des RI.

L'exécution de la simulation permettra de construire incrémentalement une carte de chaleur simulée du magasin.

Celle-ci pourra ensuite être comparée à la carte de chaleur réelle en calculant un coefficient de corrélation. L'analyse de ce coefficient permettra de vérifier et valider la simulation. La figure 4.12 synthétise cette méthodologie appliquée sur un magasin de vente.

L'intérêt d'une telle méthodologie est triple. Tout d'abord, l'étape de l'agentification, précédée par un traitement d'apprentissage, permet de disposer d'un premier résultat au niveau micro. Elle permet, en l'occurrence, de mieux comprendre les comportements des clients regroupés par type. En effet, lorsqu'un magasin de vente compte plus de 6000 clients par jour par exemple, il est difficile de suivre et repérer les intentions de consommation de tout un chacun. Il est beaucoup plus efficace de définir un nombre minimal de typologies susceptibles de caractériser les grandes catégories de la population de consommateurs.

Ensuite, la carte obtenue à l'issue de la simulation, une fois validée, permet de révéler des informations visuelles se rapportant à l'aménagement du magasin. En effet, les zones les plus visitées

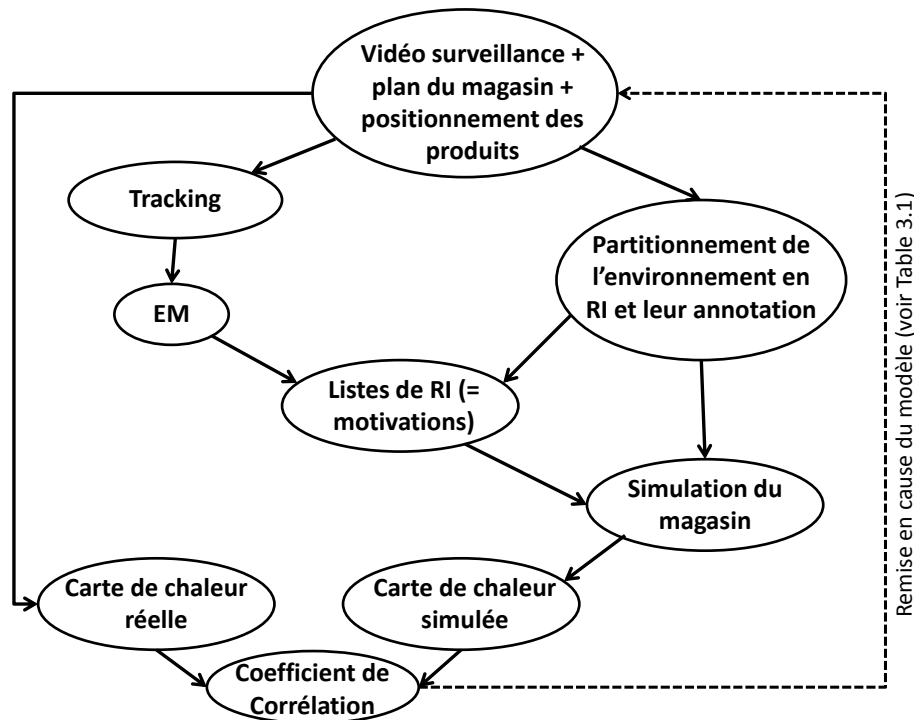


FIGURE 4.12 – La méthodologie appliquée sur un espace de vente

ont une densité accentuée, ce qui peut signifier que les produits environnants sont les plus convoités du magasin. Réciproquement, les zones les plus critiques en termes d'occupation ont une densité réduite, ce qui peut signifier que les produits environnants sont les moins convoités. Ainsi, un produit se vendant mal et placé dans une zone très dense peut être qualifié de produit non attractif. À l'inverse, un produit se vendant mal et placé dans une zone peu fréquentée peut soit être non attractif, soit pas assez mis en valeur.

Enfin, l'idée de construire une telle carte de chaleur ne s'arrête pas seulement à informer les gestionnaires du magasin de la fréquentation de certaines régions. Il est également envisageable de pouvoir modifier l'agencement (qui se traduit par un changement de la répartition des RI) et étudier son impact. Notamment, il est vraisemblable qu'on puisse prévoir les comportements potentiels des consommateurs dans le cas d'un réaménagement des linéaires. Ce qui permettrait de trouver le meilleur agencement du magasin en fonction de la visibilité de tous les produits et, donc, du chiffre d'affaires.

Retournons à notre environnement d'application représentant le hall d'entrée du bâtiment universitaire. Nous avons procédé à une permutation de certaines régions d'intérêt, notamment les RI "Coffee1", "Coffee2" et "Printer" avec les RI "Board31" et "Board32", illustrées dans la figure 4.13.

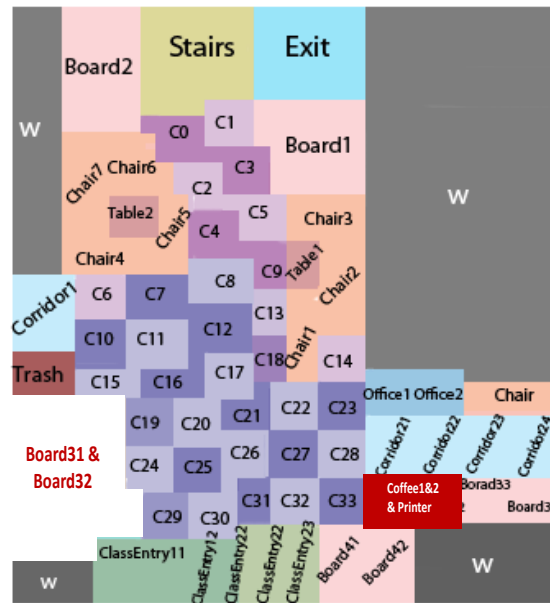
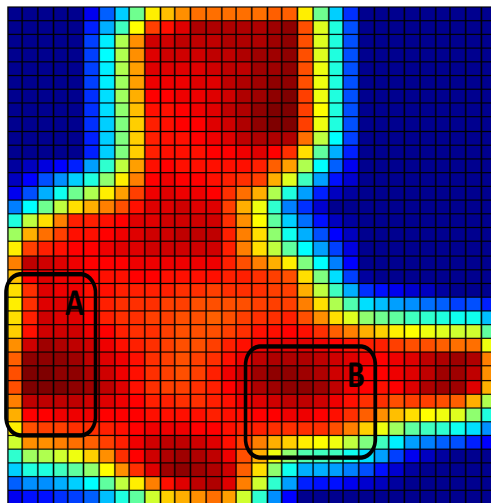
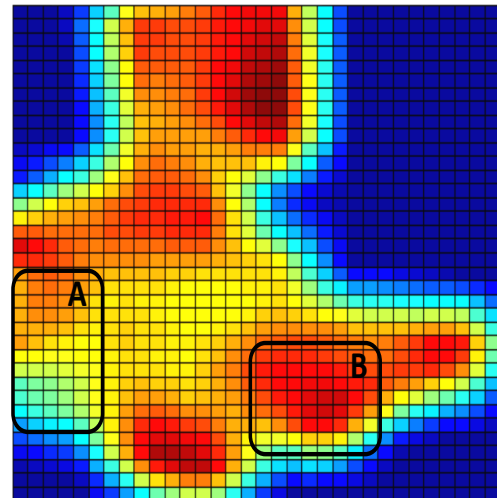


FIGURE 4.13 – Illustration de zones permutées (Board31/Board32 et Coffee1&2/Printer)



(a) Carte de chaleur simulée avant modification



(b) Carte de chaleur simulée après modification

FIGURE 4.14 – Cartes de chaleur obtenues, issues de EM, avant et après permutation des RI "Coffee1", "Coffee2" et "Printer" avec "Board31" et "Board32"

La figure 4.14 montre, à droite, la carte de chaleur résultante de la simulation suite à cette modification. Dans cette simulation, la modélisation des agents s'est faite avec l'algorithme EM. À gauche, nous retrouvons la carte de chaleur simulée sur l'agencement initial de l'environnement.

Nous remarquons, dans la figure 4.14(b)⁹, que l'ancienne zone où se trouvaient la machine à café et l'imprimante, marquée par le rectangle A, a subi un "dépeuplement" remarquable. Cela se traduit par le fait que la machine à café et l'imprimante représentent des produits fortement sollicités par les individus qui fréquentent cet environnement. La présence d'une densité importante au niveau de la zone B, représentant le nouvel emplacement de la machine à café et de l'imprimante dans la nouvelle carte de chaleur, conforte cette déduction. Inversement, cela signifie que les tableaux d'affichage, notamment "Board31" et "Board32", sont anodins et rarement consultés. Cette information passe inaperçue dans la première carte de chaleur, car les RI correspondants se trouvent à proximité d'autres RI fréquemment visités ou relatives à un flux de passage important.

Reproduire de telles modifications de l'environnement dans un magasin de vente permettrait de mettre en évidence certaines propriétés de l'agencement du magasin. L'outil reposant sur cette méthodologie de simulation multi-agents pourrait être considéré comme un outil d'aide à la décision pour les gestionnaires du magasin.

4.5 Conclusion

Ce chapitre se compose de trois parties principales. La première porte sur la conception et l'implémentation de la simulation justifiant l'organisation du travail ainsi que les choix adoptés dans les démarches suivies. La deuxième partie expose les résultats obtenus et leur évaluation. Enfin, la troisième partie apporte un éclairage sur la manière dont on pourrait transposer notre méthodologie, tout en illustrant son intérêt, sur le domaine d'application ciblé, en l'occurrence un espace de vente.

Dans un premier temps, nous avons été amenés à sélectionner une plateforme de simulation multi-agents parmi les nombreux outils existants. Notre choix s'est porté sur Repast Symphony pour sa simplicité d'accès et d'usage. À partir de ses propriétés, nous avons élaboré l'architecture de simulation sur laquelle reposent nos développements. Nous avons, également, synthétisé les différentes étapes de notre méthodologie introduite au chapitre 3 et présenté le workflow associé.

Dans un second temps, nous avons présenté les résultats obtenus par la simulation sous forme de cartes de chaleur. Ces dernières ont été générées pour différentes mises en pratique de notre méthodologie :

1. La simulation comporte des agents dont les comportements de déplacement sont appris grâce à EM à partir de l'ensemble des données observées avec un découpage de l'environnement sur un plan vue de dessus.
2. La simulation est similaire à la première, mais utilise un découpage automatique de l'environnement dans le plan caméra.
3. La simulation fait intervenir un MMC pour produire les comportements des agents sur un plan vue de dessus.
4. La simulation est similaire à la troisième, mais se produit sur le plan caméra.

9. Le code couleur est normalisé, rouge foncé représente le maximum d'affluence pour chaque carte quelque soit le niveau.

Les cartes de chaleur obtenues à partir de ces simulations ont été comparées chacune avec la carte de chaleur du monde réel correspondante. Un coefficient de corrélation a également été calculé dans chaque cas. Les résultats sont apparus réalistes au vu d'une comparaison entre les densités d'occupation de l'espace en réel et en simulé. Cependant, le comportement des agents au cours de la simulation est nettement plus cohérent lorsqu'ils sont appris avec EM plutôt qu'avec un MMC. En effet, avec EM, le comportement de déplacement des agents suit une liste de motivations représentant des régions successives dont l'ordre est interprétable. Ce qui signifie qu'on est capable d'intercepter, au cours de la simulation, l'intention de l'agent en termes de destination désirée. De plus, une des finalités des simulations est de simplifier un système réel et révéler, comme résultat, des informations pertinentes servant à mieux comprendre le système. Nous retrouvons cette particularité dans les cartes de chaleurs obtenues avec EM. L'inconvénient de cette technique (EM) est que les comportements de déplacements sont figés de telle sorte qu'un agent ne peut exécuter qu'une seule trajectoire. Quant aux MMC, il est possible d'enchaîner plusieurs comportements de déplacement accomplis par un seul agent. Cependant, les trajectoires simulées observées au niveau microscopique peuvent présenter certaines incohérences (multiples aller-retour par exemple). Toutefois, les résultats au niveau macroscopique (sous forme de cartes de chaleur) sont satisfaisants, se rapprochant fortement de la carte de chaleur réelle.

Enfin, EM et MMC apportent des résultats intéressants, et l'utilisation de l'une ou l'autre méthode dépend de la finalité de la simulation. Vu que nous envisageons de simuler un espace de vente en reproduisant les comportements de déplacement des consommateurs, nous recommandons l'utilisation d'EM plutôt que les MMC. Ce choix exprime le fait qu'il est possible d'identifier des typologies représentatives des consommateurs, caractérisés par une forte proximité comportementale, permettant de mieux comprendre leurs démarches et leurs préférences en termes de produits de consommation. De plus, les informations résultantes, au niveau global, sont plus précises et permettent une meilleure exploitation de l'environnement.

Par ailleurs, il est envisageable d'apporter des modifications à l'environnement, en terme d'agencement, afin de prévisualiser les nouveaux comportements des agents affectés par cette évolution. Nous avons, par conséquent, transposé les étapes de notre méthodologie sur ce domaine en particulier. Simuler une nouvelle organisation de l'environnement permet donc de transformer le résultat de la simulation en rapport avec cette modification. Il pourrait être envisageable d'appuyer encore plus la validation de la simulation, en modifiant l'agencement du magasin réel identiquement à la modification apportée dans la simulation. En admettant que la nouvelle carte de chaleur réelle soit analogue à la nouvelle carte simulée, la validité de l'outil de simulation à des fins de prédiction sera renforcée.

Cette méthodologie ne concerne pas uniquement le domaine de simulation d'espaces de vente. Elle pourrait traiter des domaines plus variés, notamment simuler des milieux où on a un mouvement de foule important. Par exemple, il est possible de simuler une station de métro pour étudier les flux de passage. Néanmoins, l'outil réalisé reste un prototype à perfectionner.

Conclusion générale

Dans le cadre de cette thèse, nous avons abordé une problématique originale visant la conception automatique de simulation multi-agents à partir de l'observation de comportements réels. Le cadre applicatif de cette étude concerne les espaces de vente. L'idée est de reproduire, au moyen d'une simulation multi-agents, les comportements observés des consommateurs au sein d'un magasin. L'objectif de cette thèse est donc de parvenir à modéliser et construire une simulation multi-agents en limitant le recours aux experts et ainsi réduire la part de subjectivité dans la conception du SMA.

Notre contribution porte sur trois apports distincts :

- extraction et apprentissage automatique de comportements de déplacements réels observés, traduits en listes de motivations : l'agentification ;
- conception quasi automatique de simulation multi-agents reproduisant les comportements appris ;
- évaluation et validation de la simulation en calculant un coefficient de corrélation entre le simulé et le réel.

Il existe un nombre modeste d'études qui traitent à la fois des simulations multi-agents et des comportements de consommateurs dans un point de vente. Ces études ont été exposées dans le premier chapitre et ont été scindées en deux catégories. La première traite des comportements abstraits. Autrement dit, chaque comportement est conçu sous la forme d'un modèle conceptuel orchestré par des paramètres probabilistes. La deuxième s'appuie sur des comportements réels. Il s'agit, dans ce cas, de traits de comportements résultant de l'étude de vrais consommateurs, que ce soit à partir de questionnaires exhaustifs et précis ou à partir d'équipement matériel et de logiciel. Bien que les méthodologies diffèrent d'une approche à une autre, elles ont quasiment toutes un dénominateur commun : le point de vue du concepteur qui introduit une certaine subjectivité dans la modélisation. En effet, elles reposent globalement sur des enquêtes et des formulaires ou s'appuient sur les connaissances d'experts qui sont sujets à différentes interprétations.

Comme nombre de ces travaux se base sur les simulations multi-agents, nous nous sommes orientés vers une recherche présentant des méthodes de conception de simulation multi-agents. L'idée était de trouver une approche exploitable, facilitant la construction d'une simulation multi-agents. Nous avons présenté, par conséquent, quelques méthodologies dont l'une fait participer des humains agissant sur le système, l'autre se base intégralement sur les interactions et d'autres permettent de spécifier un comportement (au niveau micro) ou de diriger le comportement global (au niveau macro). Seulement, le même inconvénient persiste, celui qui nécessite une phase d'étude empirique et donc le manque d'automatisme.

Néanmoins, une approche a particulièrement attiré notre attention. Elle consiste à combiner le Data Mining et les simulations multi-agents de manière à améliorer le fonctionnement de la simulation. Ce qui nous a amenés à réfléchir à une idée originale intégrant des techniques de Data Mining dans la conception de simulations agents. Concrètement, l'idée est d'extraire des comportements d'individus à partir de l'observation du monde réel à l'aide de techniques d'apprentissage non supervisé.

Dans cette optique, nous avons présenté dans le chapitre 2 des méthodologies existantes qui se rapportent à l'extraction automatique de connaissances à partir de séquences vidéo. Tout d'abord, nous avons décrit nos besoins en termes de types de données, comportementales et environnementales, à extraire depuis l'observation du monde réel. Nous nous sommes focalisés sur l'extraction des comportements de déplacement. Cela est pertinent, d'une part en raison de la possibilité de réaliser des suivis d'entité en mouvement sur la vidéo, et d'autre part, en raison de la connaissance engendrée. Notamment, dans un espace de vente, on est capable d'étudier à partir des trajectoires des clients, les flux de passage dans chaque linéaire, l'affluence ou l'inoccupation de certaines zones, etc. Ensuite, nous avons exposé, brièvement, différentes techniques en traitement de trajectoires. Celles-ci traitent de certains verrous relatifs au suivi des individus sur des séquences vidéo. Néanmoins, le recours à des techniques d'apprentissage apporte une meilleure modélisation des différents parcours ainsi que les principaux comportements d'individus dans la scène observée. Ceci nous a amené à exposer, enfin, les méthodologies existantes permettant l'extraction de comportements par analyse des données de trajectoires. Suite à cette analyse, nous avons retenu en particulier deux techniques qui nous ont semblé des pistes à explorer : les MMC et la technique d'EM.

A la suite de ces deux chapitres permettant de positionner notre travail, nous avons exposé au chapitre 3 notre approche que nous avons appliquée et testée sur le hall d'entrée d'un département universitaire. Il s'agit donc d'une méthodologie de conception de simulation d'agents situés dont les comportements de déplacement sont appris à partir de séquences vidéo. L'apprentissage de ces comportements est réalisé à partir d'un ensemble de données réelles observées. Les deux techniques citées auparavant, les MMC et EM, ont été testées séparément afin d'expérimenter leur potentiel à reproduire la réalité. Nous avons constaté qu'avec EM, nous sommes capables d'estimer des comportements prototypes simples sous forme de trajectoires moyennes. Quant à l'usage des MMC, permettant de rassembler tous les comportements observés en une structure unique, il est possible de combiner plusieurs comportements.

Afin de concevoir les agents, les comportements de déplacement ont été traduits en motivations suivant un partitionnement relatif à des Régions d'Intérêt (RI) de l'environnement, connu *a priori*. De ce fait, l'étape d'agentification consiste à fournir aux agents une liste d'objectifs à atteindre définissant leur parcours dans l'environnement simulé également divisé en RI correspondantes. Il est à noter qu'un minimum de connaissance *a priori* sur l'environnement est nécessaire, *a minima* le plan de l'espace à simuler (ex : magasin et rayonnages) .

Cette agentification accomplie, l'exécution de la simulation permet d'entrevoir la pertinence de EM par rapport aux MMC dans leur mise en pratique. En effet, contrairement à EM, l'application des MMC nécessite certaines opérations supplémentaires notamment sur le choix du meilleur compromis concernant le paramétrage de ce modèle. Ce compromis, portant sur le nombre d'états cachés, est choisi de façon expérimentale et dépend fortement de l'environnement concerné. Plus l'environnement

est grand et riche, plus le nombre d'états cachés est important. Dans le cas d'un espace de vente, dont l'environnement est imposant, la complexité du problème est beaucoup plus élevée.

Dans le quatrième chapitre, nous nous sommes concentrés davantage sur la simulation en termes d'implémentation et de résultats. Nous avons exposé et justifié les différents choix entrepris, relatifs à l'outil, les propriétés et l'architecture de la simulation. Nous avons ensuite défini le workflow lié à notre méthodologie décrivant chaque module traité. L'exécution de la simulation a généré des agents dont les comportements sont guidés par les listes de motivations apprises par EM ou par MMC. Nous avons remarqué, au cours de l'exécution, que les comportements des agents sont beaucoup plus réalistes et cohérents avec EM qu'avec les MMC. Lorsque les déplacements de la totalité des agents engagés dans la simulation ont été accomplis, une carte de chaleur de l'environnement simulé a été tracée. Ce résultat a permis de mettre en évidence les régions les plus visitées de l'environnement, ainsi que les régions les plus inoccupées.

Ainsi, plusieurs cartes de chaleurs ont été obtenues dans différentes configurations : utilisation d'EM par apprentissage de comportements, utilisation des MMC à différents nombres d'états cachés. Chacune de ces cartes a été analysée et comparée par rapport à la carte de chaleur réelle en se basant sur une métrique appropriée. Dans les deux cas (carte issue de EM ou MMC), nous avons montré en utilisant cette métrique que les résultats sont suffisamment similaires à la réalité pour être validés.

Toutefois, contrairement aux cartes engendrées par les MMC, les cartes issues de EM apportent une information intelligible permettant de discerner nettement l'occupation ou l'inoccupation des différentes régions d'intérêt. Etant donné que nous envisageons d'appliquer notre méthodologie sur un espace de vente, il nous semble plus adéquat d'employer la technique de EM. Avec l'information que peut révéler cette dernière, dans un environnement assez large et complexe, on serait capable de repérer les régions les plus fréquentées et les régions inexploitées. Ce qui pourrait permettre une analyse plus fine de l'agencement de l'environnement.

Enfin, nous avons pratiqué une permutation de régions dans la simulation de l'environnement étudié. Les comportements des agents étant affectés par cette modification, le comportement global de la simulation, notamment la carte de chaleur résultante, s'est également vue évoluer révélant une nouvelle connaissance. Simuler de tels changements de l'environnement permet d'effectuer certaines opérations difficilement réalisables dans le monde réel voir impossible. Supposons que nous cherchons à évaluer plusieurs agencements d'un magasin de vente, il serait beaucoup plus plausible de consulter la simulation du magasin, avec différentes configurations de l'environnement, que de déplacer à plusieurs reprises les rayons réels. Ainsi, nous souhaitons transposer notre méthodologie sur un magasin de vente réel dont le concept a été illustré à la fin du dernier chapitre.

Perspectives théoriques

L'étude menée, au cours de cette thèse, a introduit une méthodologie originale offrant un certain nombre de perspectives. Dans la mesure où, à notre connaissance, le domaine d'étude est peu exploré, il est légitime de poursuivre dans cette voie et apporter des améliorations complémentaires. Parmi les perspectives à envisager, nous citons :

- Affiner les comportements des agents. Etant donné que nous nous basons uniquement sur des comportements de déplacements pour l'identification de "modèles types", il serait intéressant de s'appuyer également sur l'analyse des gestes. Ceci permettrait de mieux comprendre l'intention des individus envers leur environnement. De plus, à partir des gestes, on est capable d'identifier certaines actions correspondant à certaines activités. Ainsi, ces identifications reposeraient sur des structures de comportements-types plus fins permettant aux agents de reproduire des comportements complets.
- Etablir des modèles de comportements plus fiables et consistants provenant de plusieurs sources de données agrégées. Dans un magasin de vente, par exemple, il serait enrichissant de prendre en considération, non seulement les données trajectoires des individus, mais également les données des tickets de caisses, les données RFID (radio-identification) lorsque celles-ci existent. D'un point de vue plus général, supposons que nous disposions de trois sources de données, il serait intéressant d'identifier des modèles de comportement à partir de l'ensemble des sources. De cette manière, il y aurait une forte probabilité d'obtenir des comportements complets, consistants et informatifs. L'agrégation des données, relatives à des entités mobiles (individus, animaux ou autres) et provenant de différentes sources, peut se faire de deux façons différentes :
 - La première façon de faire représenterait une sorte de classification hiérarchisée. Précisément, il faudrait apprendre les modèles pour chaque source l'une indépendamment de l'autre. Chaque individu est, ensuite, représenté par un vecteur de descripteurs formés à partir d'une conjonction des degrés d'appartenance aux modèles de chaque source apprise. Une classification sur l'ensemble des individus devra alors être effectuée afin d'obtenir les groupes types de comportements pour les agents (voir figure 4.15).

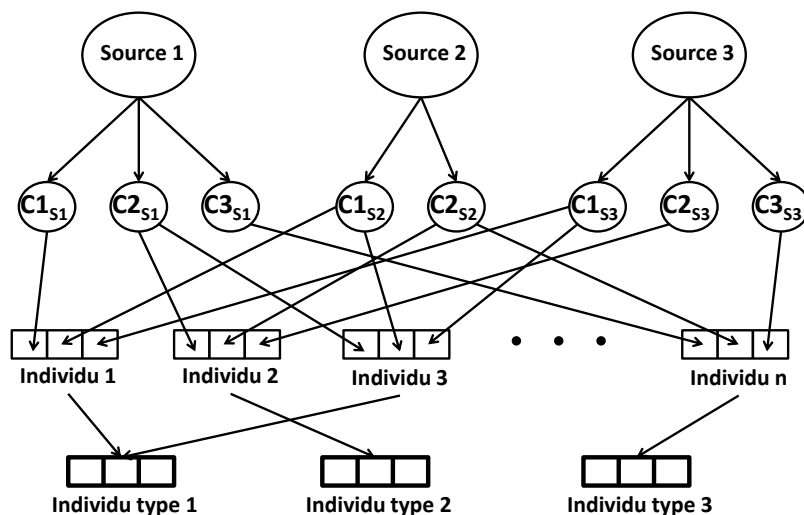


FIGURE 4.15 – Classification hiérarchique d'individus à partir de plusieurs sources de données

- Inversement, la deuxième façon de faire est de créer d'emblée des vecteurs descripteurs, correspondant à des individus observés, à partir des différentes sources selon une conjonction des données plus ou moins brutes. A partir de ces structures, il serait possible d'apprendre, au moyen d'une classification, des modèles-types formant les comportements de la population d'agents (voir figure 4.16).

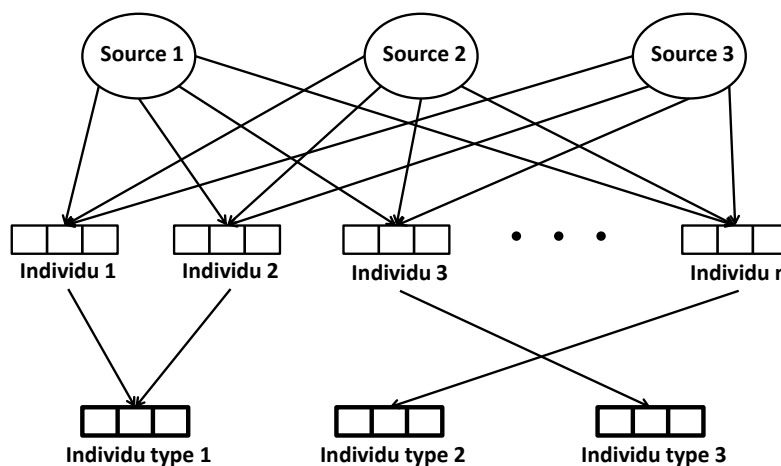


FIGURE 4.16 – Conjonction et classification d'individus à partir de plusieurs sources de données

- Identifier et simuler des comportements de groupes. Plus précisément, il s'agirait de simuler les interactions sociales. En effet, afin de mieux refléter la réalité, il serait intéressant de tenir compte de l'aspect social et par conséquent les échanges entre les individus. Ainsi, il est nécessaire d'instaurer des mécanismes de communication entre les agents.
- Réaliser un traitement d'apprentissage adaptatif par rapport aux comportements des individus afin de suivre leur évolution dans le temps. Il s'agit d'autoriser un processus d'enrichissement incrémental des modèles disponibles, à partir de données recueillies *in situ* constituant la signature d'attitudes nouvelles qu'il sera alors possible de formaliser. La représentativité du système multi-agents sera ainsi garantie par l'ajustement des motivations et des comportements. Ce qui ramène à une agentification adaptable en fonction du temps. Dans un magasin de vente, par exemple, l'intérêt d'un tel ajustement est de prendre en considération l'évolution de l'attitude des consommateurs par rapport à la saisonnalité par exemple. Lorsqu'une saison comprend des jours de fête ou une période des soldes, le comportement des consommateurs en est fortement influencé, de par son attitude et ses objectifs d'achat.

Perspectives applicatives

Après avoir élaboré et testé une méthodologie générale permettant l'apprentissage des comportements de déplacement d'individus observés dans un environnement simple, il s'agit dorénavant d'étendre le domaine applicatif et mettre en pratique cette démarche sur de vrais magasins de vente.

Il reste, toutefois, à automatiser entièrement la répartition de l'environnement, vu de dessus. Idéalement, les effets de perspective et de distorsion dus aux angles de visée des caméras de surveillance doivent être évités afin d'avoir une correspondance parfaite entre le simulé et le réel. Autrement, il serait nécessaire de trouver une fonction générique permettant d'effectuer un changement de repère, indépendamment de l'environnement étudié, ayant uniquement à disposition les propriétés de la caméra utilisée. Ainsi, il est envisageable d'automatiser le découpage de l'environnement en régions que ce soit sur la vue caméra ou la vue simulation.

Par ailleurs, il serait intéressant de peaufiner le comportement de déplacement des agents dans la simulation. Considérer un environnement continu au lieu de discret permettrait d'apporter des améliorations visuelles. Ainsi, les mouvements des agents auront une apparence beaucoup plus réaliste et le rendu beaucoup plus esthétique.

D'autre part, la méthodologie mise en œuvre dans cette thèse pourra s'appliquer à l'extraction de comportements dans un environnement non réel. Par exemple, les traces de navigation d'un internaute dans un magasin virtuel sur un site de vente en ligne pourront permettre une identification de son comportement (de déplacement virtuel) et le déclenchement de stratégies commerciales personnalisées et adaptées.

D'autres types d'applications engageant des agents situés sont vraisemblablement traitables avec notre méthodologie, notamment :

- La simulation de trafic routier où il s'agirait de reproduire des comportements de piétons à un carrefour, par exemple, où l'extraction et l'apprentissage des déplacements entraîneraient des modèles de traversée.
- La simulation d'évacuation d'urgence d'un bâtiment ouvert au public : stade, salle de concert, musée,... où il s'agirait de vérifier l'écoulement optimal de flux en fonction de sorties de secours.
- La simulation pour l'optimisation de visite : musée, château, où il s'agirait de trouver les parcours optimaux en fonction de différents critères : temps de présence minimal dans les salles, listes d'oeuvres à voir absolument, ...
- La simulation "logistique" : parc d'attraction par exemple où il s'agirait d'évaluer l'impact de l'ouverture ou de la fermeture d'une attraction sur les files d'attente des autres attractions, ...

Bibliographie

- [Alshabi et al., 2007] ALSHABI, W., RAMASWAMY, S. S., ITMI, M. et ABDULRAB, H. (2007). A cooperation mechanism in agent organization. *In SCSS (1)'07*, pages 491–496.
- [Amblard et Ferrand, 1998] AMBLARD, F. et FERRAND, N. (1998). Modélisation multi-agents de l'évolution de réseaux sociaux. *In Actes du Colloque Modèles et systèmes Multi-agents pour la gestion de l'environnement et des territoires, Clermont Ferrand*, pages 153–168.
- [Amblard et al., 2006] AMBLARD, F., ROUCHIER, J. et BOMMEL, P. (septembre 2006). Evaluation et validation de modèles multi-agents. *In Modélisation et simulation multi-agents : applications pour les Sciences de l'Homme et de la Société*, pages 103–140. Hermès Science.
- [Axtell, 1999] AXTELL, R. (1999). Why agents? on the varied motivations for agent computing in the social sciences. *Proceedings of Agent Simulation : Applications, Models, and Tools*.
- [Baqueiro et al., 2009] BAQUEIRO, O., WANG, Y. J., MCBURNEY, P. et COENEN, F. (2009). Integrating data mining and agent based modeling and simulation. *In ICDM*, pages 220–231.
- [Bar-Shalom et Fortmann, 1988] BAR-SHALOM, Y. et FORTMANN, T. (1988). *Tracking and data association*. Academic Press Professional, Inc., San Diego, CA, USA.
- [Bashir et al., 2006] BASHIR, F. I., KHOKHAR, A. A. et SCHONFELD, D. (2006). View-invariant motion trajectory-based activity classification and recognition. *Multimedia Syst.*, 12(1):45–54.
- [Bashir et al., 2007] BASHIR, F. I., KHOKHAR, A. A. et SCHONFELD, D. (2007). Object trajectory-based activity classification and recognition using hidden markov models. *IEEE Transactions on Image Processing*, 16(7):1912–1919.
- [Bernon et al., 2003] BERNON, C., CAMPS, V., GLEIZES, M. P. et PICARD, G. (2003). Designing agents' behaviors and interactions within the framework of ADELFE methodology. *In ESAW*, pages 311–327.
- [Black et Jepson, 1998] BLACK, M. J. et JEPSON, A. D. (1998). Eigenttracking : Robust matching and tracking of articulated objects using a view-based representation. *In International Journal of Computer Vision*, pages 329–342.
- [Blackman, 1986] BLACKMAN, S. (1986). *Multiple-target tracking with radar applications*. Artech House.
- [Bonnin, 2003] BONNIN, G. (2003). La mobilité du consommateur en magasin : une étude exploratoire de l'influence de l'aménagement spatial sur les stratégies d'appropriation des espaces de grande distribution. *Recherche et Application Marketing*, 18:7–29.
- [Brand et Kettner, 2000] BRAND, M. et KETTNAKER, V. (2000). Discovery and segmentation of activities in video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):844–851.

- [Bretzner et al., 2002] BRETZNER, L., LAPTEV, I. et LINDEBERG, T. (2002). Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. *In Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, pages 423–428.
- [Buechler et Smith, 1975] BUECHLER, G. et SMITH, P. (1975). A branching algorithm for discriminating and tracking multiple objects. *In IEEE Trans. Automatic Control*, volume 20, pages 101–104.
- [Buxton et Gong, 1995] BUXTON, H. et GONG, S. (1995). Advanced visual surveillance using bayesian networks. *In International Conference on Computer Vision*, pages 111–123, Cambridge, Massachusetts.
- [Chadès et al., 2002] CHADÈS, I., SCHERRER, B. et CHARPILLET, F. (2002). A heuristic approach for solving decentralized-pomdp : Assessment on the pursuit problem. *In ACM Symposium on Applied Computing - SAC'2002*, page 6 p, Madrid, Spain.
- [Di et al., 2006] DI, G., SERUGENDO, M., pierre GLEIZES, M. et KARAGEORGOS, A. (2006). Self-organisation and emergence in mas : An overview. *Informatica*, 30:45–54.
- [Drogoul et Ferber, 1994] DROGOUL, A. et FERBER, J. (1994). Multi-agent simulation as a tool for modeling societies : Application to social differentiation in ant colonies. *In Artificial Social Systems, 4th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW*, volume 830, pages 2–23. Springer.
- [Fehler et al., 2004] FEHLER, M., KLÜGL, F. et PUPPE, F. (2004). Techniques for analysis and calibration of multi-agent simulations. *In Engineering Societies in the Agents World V, 5th International Workshop, ESAW*, pages 305–321, Toulouse, France. Springer.
- [Ferber, 2007] FERBER, J. (2007). *Multi-agent Concept and Methodologies*. Agent-Based Modelling and Simulation in the Social and Human Sciences. The Bardwell Press, 6 Bardwell Road Oxford OX2 6SW.
- [Fieguth et Terzopoulos, 1997] FIEGUTH, P. et TERZOPOULOS, D. (1997). Color-based tracking of heads and other mobile objects at video frame rates. *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 21–27.
- [Ford et Moore, 1998] FORD, J. J. et MOORE, J. B. (1998). Adaptive estimation of hmm transition probabilities. *IEEE Transactions on Signal Processing*, 46(5):1374–1385.
- [Fortune, 1987] FORTUNE, S. (1987). A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2:153–174.
- [Frawley et al., 1992] FRAWLEY, W. J., PIATETSKY-SHAPIO, G. et MATHEUS, C. J. (1992). Knowledge discovery in databases : an overview. *In The American Association for Artificial Intelligence*, California.
- [Gaffney et Smyth, 1999] GAFFNEY, S. et SMYTH, P. (1999). Trajectory clustering with mixtures of regression models. *In KDD*, pages 63–72.
- [García-Ojeda et al., 2007] GARCÍA-OJEDA, J. C., DELOACH, S. A., ROBBY, OYENAN, W. H. et VALENZUELA, J. (2007). O-MaSE : A customizable approach to developing multiagent development processes. *In proc. of AOSE*, pages 1–15.
- [Gavrila et Davis, 1996] GAVRILA, D. M. et DAVIS, L. S. (1996). 3-d model-based tracking of humans in action : a multi-view approach. *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, San Francisco, USA.

- [Gechter et al., 2010] GECHTER, F., CONTET, J.-M., GRUER, P. et KOUKAM, A. (2010). Car-driving assistance using organization measurement of reactive multi-agent system. *Procedia CS*, 1(1):317–325.
- [Gelgon, 1998] GELGON, M. (1998). *Segmentation spatio-temporelle et suivi dans une séquence d'images : application à la structuration et à l'indexation de vidéo*. Thèse de doctorat, Université de Rennes 1.
- [Georgé et al., 2003] GEORGÉ, J.-P., GLEIZES, M. P. et GLIZE, P. (2003). Conception de systèmes adaptatifs à fonctionnalité émergente : la théorie amas. *Revue d'Intelligence Artificielle*, 17(4):591–626.
- [Gorelick et al., 2007] GORELICK, L., BLANK, M., SHECHTMAN, E., IRANI, M. et BASRI, R. (2007). Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253.
- [Guyot et Honiden, 2006] GUYOT, P. et HONIDEN, S. (2006). Agent-based participatory simulations : Merging multi-agent systems and role-playing games. *Journal of Artificial Societies and Social Simulation*, 9(4):8.
- [Han et al., 2007] HAN, M., XU, W., TAO, H. et GONG, Y. (2007). Multi-object trajectory tracking. In *Machine Vision and Applications, Special Issue Paper*, volume 18, pages 221–232. Springer-Verlag.
- [Hervieu et al., 2007] HERVIEU, A., BOUTHEMY, P. et CADRE, J.-P. L. (2007). A hmm-based method for recognizing dynamic video contents from trajectories. In *ICIP (4)*, pages 533–536. IEEE.
- [Hervieu et al., 2009] HERVIEU, A., BOUTHEMY, P. et LE CADRE, J.-P. (2009). Reconnaissance d'événements vidéos par l'analyse de trajectoires à l'aide de modèles de markov. *Traitement du Signal*, 26(3):187–197.
- [Hogg, 1983] HOGG, D. (1983). Model-based vision : a program to see a walking person. *Image Vision Comput.*, 1(1):5–20.
- [Howe et al., 2006] HOWE, T. R., COLLIER, N. T., NORTH, M. J., PARKER, M. T. et VOS, J. R. (2006). Containing agents : Contexts, projections, and agents. In *Proceedings of the Agent 2006 Conference on Social Agents : Results and Prospects*, University of Chicago and Argonne National Laboratory, Chicago., pages 107–113.
- [Huberman et Clearwater, 1995] HUBERMAN, B. A. et CLEARWATER, S. H. (1995). A multi-agent system for controlling building environments. In *Proceedings of the First International Conference on Multiagent Systems*, pages 171–176, San Francisco, California, USA.
- [Iglesias et al., 1997] IGLESIAS, C. A., GARIJO, M., CENTENO-GONZÁLEZ, J. et VELASCO, J. R. (1997). Analysis and design of multiagent systems using mas-common kads. In SINGH, M. P., RAO, A. S. et WOOLDRIDGE, M., éditeurs : *ATAL*, volume 1365 de *Lecture Notes in Computer Science*, pages 313–327. Springer.
- [Jeffrey, 1999] JEFFREY, G. (1999). Emergence as a construct : History and issues. *Emergence : Complexity and Organization*, 1:49–72.
- [Jennings et Bussmann, 2003] JENNINGS, N. R. et BUSSMANN, S. (2003). Agent-based control systems - why are they suited to engineering complex systems? *IEEE Control Systems Magazine*, Vol. 23, No. 3, pages 61–73.

- [Khan et Shah, 2003] KHAN, S. et SHAH, M. (2003). Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1355–1360.
- [Kim et al., 1997] KIM, J.-h., SHIM, H.-s., JUNG, M.-j., KIM, H.-s. et VADAKKEPAT, P. (1997). Cooperative multi-agent robotic systems : From the robot-soccer perspective. *In Proc. IEEE Int. Conf. Robotics and Automation*.
- [Klein, 2009] KLEIN, F. (2009). *Contrôle d'un système multi-agents réactif par modélisation et apprentissage de sa dynamique globale*. Thèse de doctorat, Ecole doctorale IAEM Lorraine.
- [Klein et al., 2009] KLEIN, F., BOURJOT, C. et CHEVRIER, V. (2009). Contribution to the Control of a MAS's Global Behaviour : Reinforcement Learning Tools. *In* ARTIKIS, A., PICARD, G. et VERCOUTER, L., éditeurs : *Engineering Societies in the Agents World IX - 9th International Workshop, ESAW 2008, Saint-Etienne, France, September 24-26, 2008, Revised Selected Papers*, volume 5485 de *Lecture Notes in Computer Science*, pages 173–190. Springer-Verlag Berlin Heidelberg.
- [Klein, 2003] KLEIN, J. (2003). breve : a 3d environment for the simulation of decentralized systems and artificial life. *In Proceedings of the eighth international conference on Artificial life, ICAL 2003*, pages 329–334, Cambridge, MA, USA. MIT Press.
- [Kubera et al., 2011] KUBERA, Y., MATHIEU, P. et SÉBASTIEN, P. (23 (3) 2011). IODA : An interaction-oriented approach for multi-agent based simulations. *Journal of Autonomous Agents and Multi-Agent Systems*, pages 303–343.
- [Ladwein, 1999] LADWEIN, R. (1999). *Le Comportement du consommateur et de l'acheteur*. Collection Gestion. Série Politique générale, finance et marketing. Economica.
- [Lipton et al., 1998] LIPTON, A., FUJIYOSHI, H. et PATIL, R. (1998). Moving target classification and tracking from real-time video. *In Proceedings of the DARPA Image Understanding Workshop*, pages 129–136, Monterey, USA.
- [Luke et al., 2005] LUKE, S., CIOFFI-REVILLA, C., PANAIT, L., SULLIVAN, K. et BALAN, G. (2005). Mason : A multi-agent simulation environment. *In Simulation : Transactions of the society for Modeling and Simulation International*, volume 82, pages 517–527.
- [Masoud et Papanikolopoulos, 2003] MASOUD, O. et PAPANIKOLOPOULOS, N. (2003). A method for human action recognition. *Image and Vision Computing*, 21:729–743.
- [Mathieu et al., 2001] MATHIEU, P., ROUTIER, J.-C. et URRO, P. (2001). Un modèle de simulation agent basé sur les interactions. *In Actes des Premières Journées Francophones sur les Modèles Formels de l'Interaction (MFI'01)*, pages 407–417.
- [McLachlan et Krishnan, 1996] MCLACHLAN, G. et KRISHNAN, T. (1996). *The EM Algorithm and Extensions*. Wiley.
- [Md Sabri et al., 2012] MD SABRI, A. Q., BOONAERT, J., LECOEUCHE, S. et MOUADDIB, E. (2012). Human action classification using surf based spatio-temporal correlated descriptors. *In IEEE Int. Conf. on Image Processing, ICIP'12*, Orlando, Florida, USA.
- [Minar et al., 1996] MINAR, N., BURKHART, R., LANGTON, C. et ASKENAZI, M. (1996). The swarm simulation system : A toolkit for building multi-agent simulations. Santa Fe Institute Working Paper " 96-06-042".

- [Mlich et Chmelar, 2008] MLICH, J. et CHMELAR, P. (2008). Trajectory classification based on hidden markov models. *In Proceedings of 18th International Conference on Computer Graphics and Vision*, pages 101–105. Lomonosov Moscow State University.
- [Moënné-Loccoz et al., 2003] MOËNNE-LOCCOZ, N., BRÉMOND, F. et THONNAT, M. (2003). Recurrent bayesian network for the recognition of human behaviors from video. *In Computer Vision Systems, Third International Conference, ICVS*, pages 68–77, Graz, Austria. Springer.
- [Morris et Trivedi, 2008] MORRIS, B. T. et TRIVEDI, M. M. (2008). Learning and classification of trajectories in dynamic scenes : A general framework for live video analysis. *In AVSS*, pages 154–161. IEEE Computer Society.
- [Morris et Trivedi, 2011] MORRIS, B. T. et TRIVEDI, M. M. (2011). Trajectory learning for activity understanding : Unsupervised, multilevel, and long-term adaptive approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(11):2287–2301.
- [Müller, 2003] MÜLLER, J.-P. (2003). Emergence of collective behaviour and problem solving. *In OMICINI, A., PETTA, P. et PITT, J., éditeurs : ESAW*, volume 3071 de *Lecture Notes in Computer Science*, pages 1–21. Springer.
- [Nascimento et al., 2005] NASCIMENTO, J., FIGUEIREDO, M. et MARQUES, J. (2005). Segmentation and classification of human activities. *British Machine Vision Conference : International Workshop on Human Activity Recognition and Modelling - HAREM 2005*, pages 79–86.
- [Nascimento et al., 2007] NASCIMENTO, J. C., FIGUEIREDO, M. A. T. et MARQUES, J. S. (2007). Learning generative models for classification-recognition of human trajectories using semi-supervised em algorithm. *In Proc. of RecPad 2007 - 13^a Conferência Portuguesa de Reconhecimento de Padrões*, Lisbon, Portugal.
- [Nascimento et al., 2010] NASCIMENTO, J. C., FIGUEIREDO, M. A. T. et MARQUES, J. S. (2010). Trajectory classification using switched dynamical hidden markov models. *IEEE Transactions on Image Processing*, 19(5):1338–1348.
- [Nguyen et Venkatesh, 2005] NGUYEN, N. et VENKATESH, S. (2005). Discovery of activity structures using the hierarchical hidden markov model. *In Proceedings of the British Machine Vision Conference*, Oxford, U.K.
- [Niu et al., 2003] NIU, W., JIAO, L., HAN, D. et WANG, Y.-F. (2003). Real-time multi-person tracking in video surveillance. *In ICICSPCM*, pages 15–18.
- [North et al., 2005] NORTH, M., HOWE, T. R., COLLIER, N. T. et VOS, J. R. (2005). The repast symphony development environment. *In Proceedings of the Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms*.
- [O'Rourke et Badler, 1980] O'ROURKE, J. et BADLER, N. (1980). Model-based image analysis of human motion using constraint propagation. *In IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 2(6), pages 522–536.
- [Parunak, 2000] PARUNAK, H. V. D. (2000). Agents in overalls : Experiences and issues in the development and deployment of industrial agent-based systems. *Int. J. Cooperative Inf. Syst.*, 9(3):209–228.
- [Parunak et al., 1998] PARUNAK, H. V. D., SAVIT, R. et RIOLO, R. L. (1998). Agent-based modeling vs. equation-based modeling : A case study and users' guide. *In MABS'98*, pages 10–25.

- [Pavón et Gómez-Sanz, 2003] PAVÓN, J. et GÓMEZ-SANZ, J. J. (2003). Agent oriented software engineering with INGENIAS. In *CEEMAS*, pages 394–403.
- [Piciarelli et al., 2008] PICIARELLI, C., MICHELONI, C. et FORESTI, G. L. (2008). Trajectory-based anomalous event detection. *IEEE Trans. Circuits Syst. Video Techn.*, 18(11):1544–1554.
- [Pierce, 2012] PIERCE, M. (2012). *Facial Expression Intelligence Scale (FEIS) : Recognizing and Interpreting Facial Expressions and Implications for Consumer Behavior*. dissertation, submitted to the faculty of the Virginia Polytechnic Institute and State University.
- [Porikli, 2004] PORIKLI, F. (2004). Trajectory pattern detection by hmm parameter space features and eigenvector clustering. In *8th European Conference on Computer Vision*.
- [Pusiol et al., 2008] PUSIOL, G., PATINO, L., BREMOND, F., THONNAT, M. et SURESH, S. (2008). Optimizing Trajectories Clustering for Activity Recognition. In *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis - MLVMA'08*, Marseille, France.
- [Rabiner et Juang, 1986] RABINER, L. R. et JUANG, B. H. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16.
- [Railsback et al., 2006] RAILSBACK, S. F., LYTINEN, S. L. et JACKSON, S. K. (2006). Agent-based simulation platforms : Review and development recommendations. *Simulation*, 82(9):609–623.
- [Randane et al., 2003] RANDANE, S., TACONET, B. et ZAHOUR, A. (2003). Apprentissage dynamique du nombre d'états d'un modèle de markov caché à observations continues : Application au tri de formulaires. *Journal Article : 19^e Colloque sur le traitement du signal et des images*, pages 1113–1116.
- [Remondino et Correndo, 2005] REMONDINO, M. et CORRENDO, G. (2005). Data mining applied to agent based simulation. In *Proceedings 19th European Conference on Modelling and Simulation Yuri Merkurjev, Richard Zobel, Eugène Kerckhoffs ECMS*.
- [Remondino et Correndo, 2006] REMONDINO, M. et CORRENDO, G. (2006). MABS validation through repeated execution and data mining analysis. *International Journal of Simulation : Systems, Science & Technology*, 7(6).
- [Ren et al., 2005] REN, X., BERG, E. C. et MALIK, J. (2005). Recovering human body configurations using pairwise constraints between parts. In *Proc. 10th Int'l. Conf. Computer Vision, Vol. 1*, pages 824–831.
- [Rerkrai et Fillbr, 2004] RERKRAI, K. et FILLBR, H. (2004). Tracking persons under partial scene occlusion using linear regression. In *8th International Student Conference on Electrical Engineering POSTER*, Prague, Faculty of Electrical Engineering, Czech Technical University.
- [Richter et al., 2006] RICHTER, U., MNIF, M., BRANKE, J., MÜLLER-SCHLOER, C. et SCHMECK, H. (2006). Towards a generic observer/controller architecture for organic computing. In HOCHBERGER, C. et LISKOWSKY, R., éditeurs : *INFORMATIK 2006 - Informatik für Menschen !*, volume 93 de *LNI*, pages 112–119. Bonner Köllen Verlag.
- [Rigoll et al., 2000] RIGOLL, G., EICKELER, S. et MÜLLER, S. (2000). Person tracking in real-world scenarios using statistical methods. In *4th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2000)*, 26-30, Grenoble, France, pages 342–347. IEEE Computer Society.
- [Rodgers et Nicewander, 1988] RODGERS, J. L. et NICEWANDER, W. A. (1988). Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42:59–66.

- [Ronfard et al., 2002] RONFARD, R., SCHMID, C. et TRIGGS, B. (2002). Learning to parse pictures of people. In *Proceedings of the 7th European Conference on Computer Vision-Part IV, ECCV '02*, pages 700–714, London, UK. Springer-Verlag.
- [Rota, 1998] ROTA, N. (1998). *Rapport de DEA : Système adaptatif pour le traitement de séquences d'images pour le suivi de personnes*. INRIA, Sophia-Antipolis.
- [Saffar et al., 2011] SAFFAR, I., DONIEC, A., BOONAERT, J. et LECOEUCHE, S. (2011). Multi-agent simulation design driven by real observations and clustering techniques. In *IEEE 23rd International Conference on Tools with Artificial Intelligence, ICTAI*, pages 555–560, Boca Raton, FL, USA.
- [Saffar et al., 2012] SAFFAR, I., DONIEC, A., BOONAERT, J. et LECOEUCHE, S. (2012). Conception de simulations multi-agents : vers une modélisation basée sur les observations des comportements réels. In *Actes de la conférence Reconnaissance des Formes et Intelligence Artificielle, RFIA*, Lyon, France.
- [Savarese et al., 2008] SAVARESE, S., DELPOZO, A., NIEBLES, J. et FEI-FEI, L. (2008). Spatial-Temporal correlators for unsupervised action classification. In *IEEE Workshop on Motion and video Computing, 2008. WMVC 2008*, pages 1–8.
- [Scherrer, 2004] SCHERRER, B. (2004). Approche connexionniste du contrôle optimal. *Journal Electronique d'Intelligence Artificielle*, 4:10 p.
- [Siebel, 2003] SIEBEL, N. (2003). *Design and Implementation of People Tracking Algorithms for Visual Surveillance Applications*. Thèse de doctorat, The University of Reading, UK.
- [Siebers et al., 2008a] SIEBERS, P.-O., AICKELIN, U., CELIA, H. et CLEGG, C. (2008a). Understanding retail productivity by simulating management practise. *CoRR*, abs/0803.1600.
- [Siebers et al., 2008b] SIEBERS, P.-O., AICKELIN, U., CELIA, H. et CLEGG, C. (2008b). Using multi-agent simulation to understand the impact of management practices on retail performance. *accédé en avril 2013 à partir de <http://red.cs.nott.ac.uk/pos/docs/pos-DMDST-MP-2008.pdf>*.
- [Simonin et Gechter, 2005] SIMONIN, O. et GECHTER, F. (2005). An environment-based methodology to design reactive multi-agent systems for problem solving. In WEYNS, D., PARUNAK, H. V. D. et MICHEL, F., éditeurs : *E4MAS*, volume 3830 de *Lecture Notes in Computer Science*, pages 32–49. Springer.
- [Spiro et al., 2010] SPIRO, I., TAYLOR, G., WILLIAMS, G. et BREGLER, C. (2010). Hands by hand : Crowd-sourced motion tracking for gesture annotation. In *IEEE CVPR Workshop on Advancing Computer Vision with Humans in the Loop (ACVHL)*.
- [Stauffer et Grimson, 1999] STAUFFER, C. et GRIMSON, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *CVPR'99*, pages 2246–2252.
- [T.Brouad et al., 1997] T.BROUAD, G. VENTURINI, M. et de BEAUVILLE, J. P. A. (1997). Apprentissage du nombre d'états d'une chaîne de markov cachée pour la reconnaissance d'images. *GRETSI, Groupe d'Etudes du Traitement du Signal et des Images*.
- [Teknomo et Gerilla, 2009] TEKNOMO, K. et GERILLA, G. P. (2009). Pedestrian static trajectory analysis of a hypermarket. In *Proceedings of the Eastern Asia Society for Transportation Studies*, volume 7.
- [Terano et al., 2009] TERANO, T., KISHIMOTO, A., TAKAHASHI, T., YAMADA, T. et TAKAHASHI, M. (2009). Agent-based in-store simulator for analyzing customer behaviors in a super-market. In *KES (2)*, pages 244–251.

- [Thome, 2007] THOME, N. (2007). *Représentations hiérarchiques et discriminantes pour la reconnaissance des formes, l'identification des personnes et l'analyse des mouvements dans les séquences d'images*. Thèse de doctorat en informatique, Université Lumière Lyon 2.
- [Tisue et Wilensky, 2004] TISUE, S. et WILENSKY, U. (2004). Netlogo : A simple environment for modeling complexity. *In International Conference on Complex Systems*, pages 16–21.
- [Trakulpipat et Sinthupinyo, 2008] TRAKULPIPAT, C. et SINTHUPINYO, S. (2008). Designing market stall layout by agent-based simulation. *In International Multiconference of Engineers and Computer Scientists*, volume 1, pages 35–39.
- [Tuyls et al., 2006] TUYLS, K., HOEN, P. J., VERBEECK, K. et SEN, S., éditeurs (2006). *Learning and Adaption in Multi-Agent Systems, First International Workshop, LAMAS 2005, Utrecht, The Netherlands, July 25, 2005, Revised Selected Papers*, volume 3898 de *Lecture Notes in Computer Science*. Springer.
- [Wiliem et al., 2009] WILIEM, A., MADASU, V. K., BOLES, W. W. et YARLAGADDA, P. K. D. V. (2009). A context-based approach for detecting suspicious behaviours. *In DICTA*, pages 146–153. IEEE Computer Society.
- [Witten et al., 1999] WITTEN, I. H., FRANK, E., TRIGG, L., HALL, M., HOLMES, G. et CUNNINGHAM, S. J. (1999). Weka : Practical machine learning tools and techniques with java implementations. *Artificial Neural Networks and Expert Systems - ANNES*.
- [Wolf et Holvoet, 2005] WOLF, T. D. et HOLVOET, T. (2005). Emergence versus self-organisation : Different concepts but promising when combined. *In Engineering Self-Organising Systems*, volume 3464, pages 1–15. Springer-Verlag.
- [Wooldridge et al., 2000] WOOLDRIDGE, M., JENNINGS, N. R. et KINNY, D. (2000). The Gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312.
- [Yamato et al., 1992] YAMATO, J., OHYA, J. et ISHII, K. (1992). Recognizing human action in time-sequential images using hidden markov model. *In IEEE Computer Society Conference on In Computer Vision and Pattern Recognition*, pages 379–385.
- [Zhao. et Huang, 2002] ZHAO., H.-X. et HUANG, Y.-S. (2002). Real-time multiple-person tracking system. *In International Conference on Pattern Recognition*, volume 2, pages 897–900.

Annexe A

Proposition d'automatisation du découpage de l'environnement

Le partitionnement de l'environnement en régions et sa labellisation ont été effectués, dans un premier temps, manuellement (en prenant en compte certaines connaissances sur celui-ci). Comme nous visons à automatiser autant que possible la conception de la simulation, nous avons étudié la possibilité de faciliter cette étape liée au découpage de l'environnement d'étude.

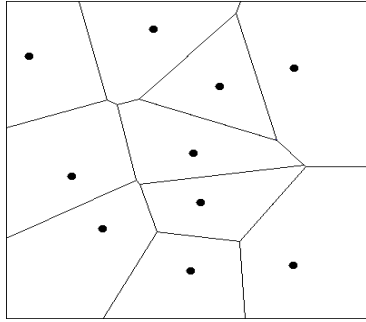
Nous envisageons ici de réaliser ce découpage en nous basant exclusivement sur les données à disposition, notamment les images capturées par la caméra et les trajectoires extraites de celles-ci. D'un point de vue algorithmique, nous nous basons sur le principe de Voronoï [Fortune, 1987], présenté brièvement ci-après.

Diagramme de Voronoï

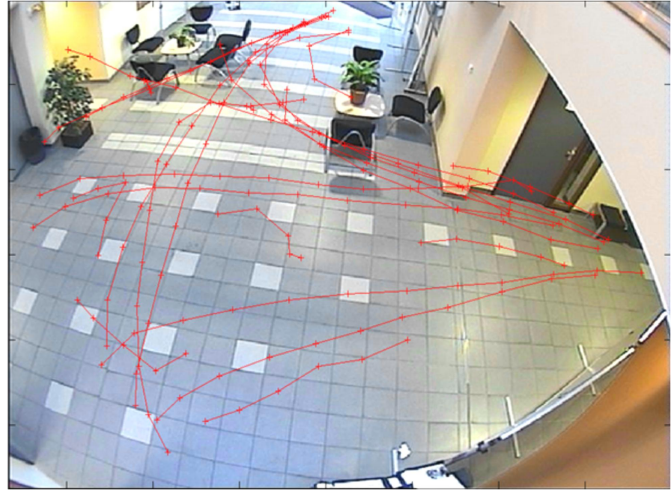
Un diagramme de Voronoï, également appelé partition de Voronoï, est un type de partition d'un espace euclidien. Soit E un espace euclidien et S un ensemble fini de n points de E appelés sites de Voronoï. On appelle région de Voronoï ou cellule de Voronoï associée à un élément p de S l'ensemble des points de E qui sont plus proches de p que de tous les autres éléments de S , la proximité étant définie par une distance. Les régions de Voronoï formées sont des polytopes convexes, en tant qu'intersections de demi-espaces. La figure A.1(a) représente le diagramme de Voronoï d'un ensemble de 10 sites dans le plan, la distance utilisée étant la distance euclidienne.

Réduction du nombre de sites

L'approche proposée se base exclusivement sur les trajectoires, et plus précisément sur les trajectoires moyennes obtenues par le clustering. Elle utilise un diagramme de Voronoï à partir des points de la trajectoire moyenne de chaque cluster. Le nombre de régions dépend du nombre de clusters et de la taille de chaque trajectoire moyenne. Ainsi, considérant 24 clusters et des trajectoires moyennes de 50 points, le nombre de régions est de 1200, ce qui est un découpage beaucoup trop fin. Il est donc nécessaire de ré-échantillonner les trajectoires moyennes (figure A.1(b)) et de regrouper les points de trajectoire proches (figure A.2). Pour chaque ensemble de points proches (i.e. dont les distances deux à deux sont inférieures à un seuil donné), il est possible de remplacer ceux-ci par un nouveau point : leur isobarycentre.



(a) Diagramme de Voronoï d'un ensemble de 10 sites dans le plan, en distance euclidienne



(b) Les 24 trajectoires moyennes après le nouvel échantillonnage

FIGURE A.1 – Le principe de Voronoï

Ainsi, après interpolation linéaire sur les trajectoires moyennes de notre exemple, nous obtenons 188 points. Après fusion des points proches, nous passons de 188 points à 70 points ce qui dans notre exemple est raisonnable au regard de la taille de l'environnement considéré.

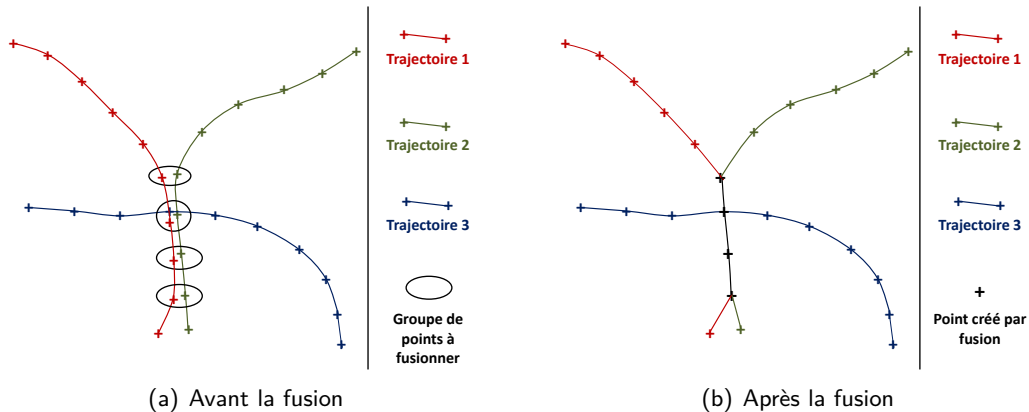


FIGURE A.2 – Exemple de fusion avec 3 trajectoires

Application du diagramme de Voronoï discret

Comme nous faisons abstraction du changement de repère permettant de passer en vue de dessus pour la simulation, l'environnement utilisé est représenté par une grille de dimension proportionnelle à celle de l'image, c'est-à-dire 57 cases en largeur et 42 en hauteur. Avec une grille de cette taille, et 70 sites de Voronoï, l'erreur d'approximation du diagramme est faible. On obtient, au final, la partition de l'espace représentée par la figure A.3.



FIGURE A.3 – Partition de l'environnement utilisant le diagramme de Voronoï discret

Limitation par l'écart-type

Le diagramme de Voronoï crée une partition de l'ensemble de la zone de simulation. Cependant, une partie non négligeable de l'environnement n'est absolument pas fréquentée, notamment les murs visibles de part et d'autre du hall. Il n'est donc pas utile de partitionner ces régions, et l'absence de sites dans ces régions crée un allongement sur les bords de l'image des régions de Voronoï en périphérie.

Pour pallier ce problème, les zones sont bornées en utilisant les écarts-types associés à chaque site. En effet, chaque site correspond à un point de la trajectoire moyenne de l'un des clusters, ou à la fusion de plusieurs d'entre eux, et par conséquent deux écarts-types associés, l'un en abscisse, l'autre en ordonnée. Si le site a été obtenu par fusion, les écarts-types sont la moyenne des écarts-type des points fusionnés. Ainsi, on définit pour chaque site de coordonnées $(\bar{x}, \bar{y})$ ayant pour écarts-types associés σ_x en abscisse et σ_y en ordonnée, le rectangle limitant contenant l'ensemble des cases de coordonnées (x, y) pour lesquelles on a :

$$\begin{cases} \bar{x} - \sigma_x \leq x \leq \bar{x} + \sigma_x \\ \bar{y} - \sigma_y \leq y \leq \bar{y} + \sigma_y \end{cases}$$

Si on supprime des régions de Voronoï, précédemment créées, les cases de la grille qui n'appartiennent pas au rectangle limitant correspondant, on réduit la taille des cellules en bord de mur. Nous obtenons, au final, la partition de l'espace représentée par la figure A.3.

Malgré l'aspect constructif et automatique de la méthode présentée, on est, néanmoins, contraint à considérer l'environnement sur le plan caméra. Le changement de repère vers le plan du sol n'a

pas été réalisé car la caméra utilisée ne nous permet pas de le faire. En effet, en sélectionnant plusieurs points dans la vue caméra et les points images dans la vue simulation, il est possible de trouver la matrice de l'application qui transforme les points du repère caméra en des points dans le repère simulation. Néanmoins, l'application linéaire obtenue par cette méthode possédait une erreur beaucoup trop importante pour être exploitable.

Résumé : La conception d'outils de simulation capables de reproduire la dynamique et l'évolution de phénomènes complexes réels est une tâche difficile. La modélisation de ces phénomènes par des approches analytiques est souvent impossible, obligeant le concepteur à s'orienter vers des approches comportementales.

Dans ce contexte, les simulations multi-agents représentent aujourd'hui une alternative crédible aux simulations classiques. Elles restent cependant délicates à mettre en œuvre. En effet, le concepteur de la simulation doit être capable de transcrire en comportement d'agents la dynamique du phénomène qu'il observe. Cette étape requiert généralement les compétences d'un spécialiste possédant une certaine expertise du phénomène à simuler.

Dans cette thèse, nous proposons une manière originale de traiter l'observation de comportements réels à simuler, sans avoir recours à l'aide d'un expert. Il s'agit de s'appuyer sur des techniques d'apprentissage non supervisé pour identifier et extraire des comportements et ainsi faciliter l'*agentification* de la simulation.

Notre approche constitue, de ce fait, un pas vers la conception automatique de simulations multi-agents reproduisant des phénomènes observables. Cette approche est motivée par un cadre applicatif visant la simulation de comportements de clients à l'intérieur d'un espace de vente.

Mots clés : Simulation Multi-Agents, Fouille de données, Modélisation de comportements, Conception automatique d'agents, Techniques de segmentation, Modèles de Markov Cachés, techniques d'Estimation-Maximisation, Agentification.

Abstract: The design of simulation tools, which are able to reproduce the dynamics and evolution of complex real phenomena, is hard. Modeling these phenomena by analytical approaches is often unsuitable, forcing the designer to turn towards behavioral approaches.

In this context, multi-agent simulations are now a credible alternative to the classical simulations. However, they remain difficult to implement. In fact, the designer of the simulation must be able to transcribe the dynamic of the phenomenon being observed in agents behavior. This step usually requires the skills of a specialist with some expertise in the phenomenon to be simulated.

In this thesis, we propose a novel way to treat observing real behaviors to simulate, without resorting to the help of an expert. It is relying on unsupervised learning techniques to identify and extract behavior and facilitate the agentification.

Our approach is, therefore, a step towards the automatic design of multi-agent simulations reproducing observable phenomena. This approach is motivated by an application context aiming the simulation of customers behavior within a retail space.

Keywords: Multiagents Simulation, Data Mining, Behavior modelling, Automatic Agent Design, Clustering technique, Hidden Markov Model, Estimation-Maximisation technics, Agentification.