Caractérisation et Reconnaissance de Gestes dans des vidéos à l'aide de Modèles Markoviens

Selma Belgacem

Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes
École Doctorale SPMII, Université de Rouen
U.F.R des Sciences et Techniques

THÈSE DE DOCTORAT

pour l'obtention du grade de
DOCTEUR DE L'UNIVERSITÉ DE ROUEN

Discipline : Informatique
Spécialité : Vision par ordinateur

Caractérisation et Reconnaissance de Gestes dans des vidéos à l'aide de Modèles Markoviens

---

Réalisée par

Selma BELGACEM

Soutenue le 25 Juin 2014 devant le jury composé de :

M. Eric ANQUETIL	Professeur des Universités	INSA Rennes	Rapporteur
M. Vlad Stefan BARBU	Maître de Conférences	Université de Rouen	Invité
M. Clément CHATELAIN	Maître de Conférences	INSA Rouen	Co-encadrant
M. Patrice DALLE	Professeur des Universités	Université Paul Sabatier	Rapporteur
M. Thierry PAQUET	Professeur des Universités	Université de Rouen	Directeur de thèse
Mme. Su RUAN	Professeur des Universités	Université de Rouen	Examinateur
Mme. Nicole VINCENT	Professeur des Universités	Université Paris Descartes	Examinateur

Version du 4 juin 2014


بسم الله الرحمن الرحيم
بِسْمِ اللهِ الرَّحْمَنِ الرَّحِيمِ
Bismillâhi-rahmâni-rahîm
Au Nom de Dieu le Miséricordieux,
le Très Miséricordieux

Remerciements

Il m'est particulièrement agréable, avant de présenter ce travail de thèse, d'exprimer toute ma gratitude envers les personnes qui, de prés ou de loin, m'ont apporté leur soutien.

J'adresse tout d'abord ma plus grande reconnaissance à mon directeur de thèse, Thierry Paquet, et mon co-encadrant Clément Chatelain pour leur excellent encadrement et pour le suivi attentif et continu de ce travail de thèse. Je les remercie pour m'avoir donné leur confiance, et veillé à la rigueur scientifique de ce travail. Je tiens à remercier tout particulièrement Thierry Paquet pour son aide à surmonter les difficultés financières confrontées durant ces années de thèse.

J'exprime ma profonde gratitude aux membres de jury de ma soutenance de thèse : Eric Anquetil, Vlad Stephan Barbu, Patrice Dalle, Su Ruan et Nicole Vincent, pour l'intérêt porté à mes travaux. Je tiens à remercier tout particulièrement Eric Anquetil et Patrice Dalle pour avoir accepté de rapporter ces travaux dans ces brefs délais.

Je tiens également à remercier tous les membres de l'équipe « Document et Apprentissage » pour leur coopération et pour l'ambiance amicale qu'ils m'ont procuré pendant ces années de thèse. Je remercie également tout le personnel administratif du Laboratoire LITIS pour son efficacité et sa disponibilité.

J'exprime une vive reconnaissance à Achraf Ben Hamadou et Walid Mahdi, membres du laboratoire MIRACL installé à Sfax en Tunisie, pour leur coopération enrichissante pendant les travaux de la première partie de cette thèse.

Je tiens à exprimer une grande reconnaissance à mes proches et mes amies en France et en Tunisie pour leur soutien et leur encouragement. Je remercie tout d'abord mes chers parents qui n'ont jamais cessé de me fournir les moyens pour avancer dans mes études et aller jusqu'au bout. Je remercie également ma soeur, mon frère et ma grande famille pour leur encouragement continu. Je tiens à remercier également mon fiancé et sa famille pour leur soutien sans cesse. Je remercie également la famille Chriaa et la famille Ammar de Rouen qui m'ont souvent accueilli et soutenu pendant ces années de thèse passées à Rouen. Je remercie enfin toutes mes amies : Ibtissem, Saida, Fatiha, Imen, Maroua, Ouardia, Boutheina..., pour leur générosité et leur assistance pendant ces années de thèse.

Résumé

Cette thèse concerne l'analyse de gestes, et plus particulièrement la caractérisation et la reconnaissance de gestes. L'analyse des données gestuelles est un axe de recherche impliquant les domaines de la communication Homme-Machine, de gestion des documents vidéo et du traitement du signal.

La contribution principale de notre travail de thèse est l'étude, la conception et la mise en œuvre, d'un système markovien hybride pour la reconnaissance des données séquentielles. La tâche de reconnaissance combine classiquement deux tâches : la segmentation et la classification. Le modèle hybride proposé combine ainsi la capacité de modélisation et de segmentation des Modèles de Markov Cachés et la capacité de discrimination locale des Champs Aléatoires Conditionnels. Nous avons appliqué ce système hybride à la reconnaissance des séquences de gestes dans des vidéos, dans le cadre applicatif de l'apprentissage avec un seul exemple. Les bonnes performances de reconnaissance obtenues dans le contexte de la compétition `ChaLearn` montrent l'intérêt de l'approche proposée pour l'apprentissage avec peu de données.

La tâche de reconnaissance nécessite une étape de caractérisation des données. Dans le cadre de la caractérisation des gestes, nous proposons deux contributions. La première contribution est une amélioration du suivi local de la main dominante dans un geste avec les filtres particulaires. Cette amélioration est basée principalement sur une pénalisation issue des flots optiques de l'estimateur, et une génération automatique d'un vocabulaire de référence. La deuxième contribution est la proposition d'une méthode de caractérisation globale du geste que nous appelons une « signature du geste ». La signature du geste décrit la localisation, la vélocité et l'orientation du mouvement global dans un geste en combinant des informations de vitesse calculée avec les flots optiques.

Mots-clés : reconnaissance de gestes, vidéos, système hybride, modèle de Markov cachés, champ aléatoire conditionnel, caractérisation de gestes, filtres particulaires, flots optiques.

Abstract

This PHD thesis concerns the analysis of gestures, especially the characterization and the recognition of gestures. The analysis of gestural data is a research field which involves Human-Machine communication, video management and signal processing fields.

The main contribution of this PHD thesis is the design and implementation of a hybrid Markov system for sequential data recognition. The recognition task typically combines two tasks : segmentation and classification. Therefore, the proposed hybrid model combines the ability of modeling and segmentation of Hidden Markov Models and the ability of local discrimination of Conditional Random Fields. We applied this hybrid system to the recognition of gesture sequences in videos in the context of one-shot-learning. The interesting recognition performances achieved in the context of the competition of `ChaLearn` show the advantage of the proposed approach for the context of learning with few examples.

The recognition task requires a step of data characterization. In the context of gesture characterization, we propose two contributions. The first contribution is an improvement of local tracking of the dominant hand in a gesture with particle filters. This improvement is mainly based on a penalisation, computed with optical flow method, of the estimator and an automatic vocabulary reference generation. The second contribution is a method of global characterization of a gesture that we call the "gesture signature". The gesture signature describes the location, velocity and orientation of the global movement in a gesture combining velocity information calculated with optical flow method.

Keywords : gesture recognition , video, hybrid system , hidden Markov model, conditional random field, gesture characterization, particle filters, optical flow.

Table des matières

Chapitre 1

Introduction Générale

Table des matières

Le sujet de cette thèse concerne la caractérisation et la reconnaissance des gestes dans des vidéos en appliquant des modèles markoviens. Cette étude est applicable dans plusieurs contextes d'interaction Homme-Machine ou de gestion des documents de type vidéo. Dans cette introduction générale, nous commençons par expliciter ces différents contextes et les besoins qui en découlent. Ensuite, nous précisons la problématique soulevée dans cette étude. Puis, nous donnons un aperçu sur les techniques markoviennes qui sont la méthode à l'état de l'art répondant à cette problématique. Nous terminons ce chapitre par l'explication de l'organisation du manuscrit et les contributions réalisées durant cette étude.

1.1 Cadre général

Un geste est un mouvement corporel humain de durée courte, de l'ordre de quelques secondes, réalisé essentiellement avec les bras. Il permet de communiquer avec les machines qui l'interprètent à l'aide des techniques de vision par ordinateur. Pour ces machines, un geste représente une donnée séquentielle nécessitant un support de stockage dynamique comme la vidéo ou le signal selon le mode

de capture de l'information. Ainsi, l'analyse des données gestuelles est un axe de recherche lié à des domaines de communication Homme-Machine, de gestion des documents vidéo et du traitement du signal.

Le domaine d'analyse des gestes inclut plusieurs thèmes : la caractérisation, le suivi, la reconnaissance, la segmentation, la détection, ce qu'on appelle également le « *spotting* » de gestes qui consiste à détecter et reconnaitre certains gestes spécifiques dans une vidéo, etc. Dans le cadre de notre étude, nous nous intéressons à la caractérisation et la reconnaissance des gestes. La caractérisation des gestes consiste à extraire, à partir des données, des informations distinguant les classes de gestes. La caractérisation des gestes est une étape indispensable pour la reconnaissance des gestes. La reconnaissance des gestes consiste à identifier les gestes par la machine. C'est une tâche nécessaire pour les applications de communication Homme-Machine.

En effet, l'évolution de la technologie nécessite le développement de techniques efficaces de communication entre l'Homme et les machines. Pour simplifier l'interaction Homme-Machine, les concepteurs des machines ont tendance à réduire le nombre de périphériques d'entrée et les remplacer par une acquisition et une reconnaissance directe des gestes. Cette constatation s'applique aux jeux vidéos, au contrôle des robots et à la traduction des langues gestuelles [54].

Nous citons comme premier exemple de langues gestuelles, la langue des signes [128]. La langue des signes est la langue principale adoptée par les personnes sourdes et muettes et présente un besoin de traduction important. La traduction automatique de cette langue gestuelle vers une langue écrite ou parlée à travers des applications informatiques facilite la communication entre les personnes à difficultés auditives et le reste de la société. La traduction automatique permet de résoudre le problème de disponibilité des spécialistes de traductions de la langue des signes auprès des services administratifs ou dans les cas d'urgence. Certains travaux [36, 141, 142] ont été réalisés dans le but de construire des systèmes de traduction de la langues des signes. La traduction des discours en langue des signes est réalisée avec des techniques de reconnaissance de signes combinées à des connaissances linguistiques de haut niveau comme les règles grammaticales. Cette langue se compose d'un ensemble de signes articulés d'une manière complexe. Chaque signe peut être représenté par une forme instantanée ou par un ou plusieurs gestes. Les gestes sont réalisés principalement par les mains et le visage. La langue des signes est caractérisée par une main dominante qui illustre les formes et les mouvements dominants dans la plupart des signes. La trajectoire et la forme de cette main dominante peuvent jouer un rôle principal dans la détermination du signe effectué. Dans la première partie de notre travail, nous nous sommes intéressés à la caractérisation et au suivi de gestes en prenant pour exemple des vidéos en langue des signes.

La langue des plongeurs [1] est considérée également comme une langue gestuelle. Dans la pratique, les chercheurs souhaitent appliquer la reconnaissance automatique de la langue des plongeurs pour le contrôle des robots sous-marins [2]. Récemment, le matériel sous-marin est devenu de plus en plus sophistiqué, autonome et doté d'une intelligence considérable comme le robot « Little Hercules » [3]. Concevoir des robots sous-marins avec une intelligence élevée répond au besoin d'explorer les zones sous-marines dangereuses. Ces robots seront ainsi des outils intelligents d'exploration complémentaires aux plongeurs. Afin de faciliter la tâche du plongeur, le contrôle manuel du robot sous-marin peut être remplacé par un contrôle gestuel-visuel. Le plongeur communique ainsi avec le robot à travers la langue des plongeurs et le robot réagit en contre partie grâce aux techniques de vision par ordinateur, de suivi et de reconnaissance des gestes. Avec de telles capacités le robot répond au besoin d'assurer la sécurité du plongeur en cas de danger.

Un autre contexte de communication Homme-Machine largement étudié est le contexte des Jeux vidéo. Un des dispositifs de jeux vidéos récemment développé et largement utilisé est le système Kinect [4]. Grâce aux techniques de vision par ordinateur, il permet le contrôle des jeux avec les gestes humains sans périphérique intermédiaire. Ce dispositif permet de construire des bases de données de l'activité humaines sous différentes modalités imagerie couleur et carte de profondeur de la scène. Fournissant des bases de données de ce type, une compétition annuelle, éditée par l'organisation `ChaLearn` [54], est organisée depuis l'année 2011 pour stimuler les recherches dans le domaine de la reconnaissance de l'activité humaine.

Tous ces systèmes de capture de gestes fournissent des données séquentielles enregistrées dans des documents vidéos ou sous forme de signaux selon le mode de capture de l'information. Étant donné la simplicité et la fréquence d'utilisation de la caméra vidéo dans la pratique, nous nous intéressons aux gestes stockés dans des documents vidéos avec une caméra fixe.

La vidéo est un document numérique de stockage de l'information audiovisuelle. Les facilités d'acquisition numérique et le besoin de mémorisation et de partage massif d'informations multimédia sur les réseaux informatiques génèrent un volume très important de ce type de document nécessitant des outils de gestion automatique fiables et rapides. En général, la gestion des documents repose sur deux processus principaux ; le processus de recherche des documents nécessitant des techniques d'indexation ou de caractérisation des documents et le processus d'extraction de l'information nécessitant les techniques de reconnaissance d'évènements. En ce qui concerne les documents de type vidéo, ces techniques sont en

1. http ://www.plongeemarseille.fr/plonger-marseille/le-langage-des-plongeurs
2. exemple : le projet européens COMMAR : http ://cordis.europa.eu/result/brief/rcn/11882_fr.html
3. http ://oceanexplorer.noaa.gov/okeanos/edu/collection/media/hdwe-URintro.pdf
4. http ://www.xbox.com/fr-FR/Kinect

cours de développement [23, 104]. Par exemple, le moteur de recherche `Google` met à la disposition de l'utilisateur différents type de mots clés pour la recherche de documents tels que le texte, la parole ou l'image. La recherche dans des vidéos n'est pas encore mise en œuvre. Les techniques de recherche et d'indexation nécessitent des méthodes de reconnaissance des éléments clés. Ces éléments clés sont souvent représentés par des gestes. Ainsi, le processus d'indexation et de recherche de vidéos contenant des activités humaines nécessite généralement des techniques de reconnaissance des gestes. Nous citons comme deuxième exemple l'institut national de l'audiovisuel (INA) qui intègre parmi ses axes de recherche la recherche de gestes dans des vidéos [5].

Ces contextes applicatifs présentent un besoin de développer des systèmes de reconnaissances de gestes. Un même système de reconnaissance est adaptable à toutes ces applications. C'est dans cette orientation que notre travail de thèse s'intègre. Nous proposons un système générique de reconnaissance de gestes reposant sur des modèles Markoviens. Dans la prochaine section, nous présentons les difficultés auxquelles un système de reconnaissance de gestes peut être confronté.

1.2 Problématique de la reconnaissance de gestes

La reconnaissance de gestes est une tâche d'identification des données gestuelles. Dans le cas de la reconnaissance de discours gestuels, qui sont des ensembles articulés de gestes, la reconnaissance combine deux tâches : la segmentation et la classification. Comme l'affirme Sayre [121], la segmentation et la classification sont deux tâches qui doivent être réalisées simultanément. La tâche de segmentation doit déterminer les limites des gestes sur la séquence. La tâche de classification doit attribuer à chaque sous-séquence une étiquette appartenant à un vocabulaire de gestes donné. La tâche de classification doit intégrer également des connaissances *a priori* sur les données telles que le vocabulaire des gestes, les durées des gestes, l'environnement d'enregistrement, etc. La difficulté principale que comporte l'étape de segmentation est la variabilité de la durée des instances d'un même geste. L'étape de classification est sujette à la variabilité des caractéristiques des instances d'un même geste.

Un geste est un ensemble de mouvements effectués essentiellement avec les mains. Il peut être représenté dans un espace simplifié tridimensionnel constitué de sa projection bidimensionnelle et de sa variation dans le temps. La problématique principale qui se pose par conséquent est la modélisation de cette variation tridimensionnelle et quel est le système capable de reconnaître ce type de données en prenant en compte tous les paramètres spatiotemporels.

Les durées des instances d'un geste ne sont pas nécessairement identiques. Cet variabilité est due principalement au changement de vitesse de réalisation du

5. http ://www.institut-national-audiovisuel.fr/nous-connaitre/entreprise/index.html

geste par une même personne ou par des personnes différentes. Ainsi, le système de reconnaissance doit être capable de modéliser cette élasticité et d'identifier le geste en dépit de la variation de durée de ses instances.

Outre la robustesse à l'élasticité des instances du geste, le système de reconnaissance doit également être robuste aux variabilités liées à l'environnement d'enregistrement. En effet, les conditions d'enregistrement ne sont généralement pas identiques entre deux séquences représentant le même geste. Nous pouvons ainsi observer des changements de luminosité, des fonds variables contenant plusieurs couleurs ou plusieurs objets et des vêtements aux formes et aux couleurs variables.

Ainsi, le système de reconnaissance doit gérer ces variabilités à travers une caractérisation robuste du geste. Si les instances des caractéristiques du même geste présentent un changement remarquable entre deux vidéos, il devient difficile pour le système de reconnaissance d'identifier le même geste sur les deux vidéos. Ces caractéristiques doivent ainsi minimiser la variabilité intra-classes et maximiser la variabilité inter-classes.

Parmi les techniques de modélisation des données séquentielles aidant à résoudre ces difficultés nous utilisons les méthodes statistiques markoviennes que nous allons présenter dans la section suivante.

1.3 Modèles markoviens

Les modèles markoviens sont des modèles largement appliqués à la reconnaissance et la segmentation de données séquentielles. Ils modélisent les dépendances temporelles des séquences. Pour cela, ils sont basés sur l'hypothèse de Markov qui simplifie les dépendances entre les états temporels de la séquence en ne modélisant que les dépendances à court terme et en omettant les dépendances sur des durées plus longues.

Bien que les systèmes markoviens simplifient les dépendances entre les différents éléments du modèle, ils fournissent une structuration globale de la donnée permettant de prendre en compte des connaissances contextuelles de haut niveau comme la grammaires des données linguistiques. Il existe également des systèmes markoviens, tels que les champs aléatoires conditionnels [143], qui prennent en compte les dépendances locales entre chaque observation et son contexte. Cette structuration globale et locale facilite la tâche de reconnaissance des données séquentielles, notamment des gestes.

Le premier défi évoqué dans la problématique est la modélisation de la durée variable des instances des gestes. Les modèles de Markov, tels que les modèles de Markov cachés [109], sont des classifieurs dynamiques capables de modéliser des signaux de durée variable. Ils ont la capacité de prendre en compte cette variabilité à l'aide des probabilités d'auto-transition des états du modèle. Ces capacités de

structuration globale et de gestion de la variabilité de la durée permettent aux modèles de Markov cachés de réaliser la tâche de segmentation.

Le deuxième type de variabilité qui présente une difficulté pour les systèmes de reconnaissance de gestes est la variabilité liée aux conditions d'enregistrements des données vidéos telles que la variabilité de la luminosité, du fond, des vêtements de l'acteur... Les modèles markoviens possèdent une certaine flexibilité leur permettant de résister à ces variabilités grâce à leur modèle probabiliste d'attache aux données.

D'autre part, comme nous l'avons expliqué précédemment, la reconnaissance de gestes nécessite une étape préliminaire de caractérisation des gestes. Nous considérons que le suivi du geste dans une vidéo constitue une méthode de caractérisation locale des gestes qui permet de caractériser le geste par un ensemble de points d'intérêt spécifiques. En effet, le suivi des éléments générant le geste, comme les mains, permet d'extraire des trajectoires décrivant les mouvements dominant du geste. Les modèles markoviens, tels que les filtres particulaires [49], sont également capable de suivre les gestes. Notons que les modèles markoviens de suivi nécessitent eux-même d'autres méthodes de caractérisation de la cible qui soient robustes aux variabilités liées à l'environnement d'enregistrement.

En conclusion, avec toutes ces spécificités, les modèles markoviens sont *a priori* adaptés à accomplir la tâche de caractérisation et de reconnaissance des gestes avec certaines limites que nous allons étudier et tenter de résoudre dans ce travail de thèse.

1.4 Contributions

Certains modèles markoviens, comme les champs aléatoires conditionnels (les `CAC`) [143], sont *a priori* adaptés au problème de classification et d'autre modèles markoviens, comme les modèles de Markov cachés (les `MMC`) [109], sont *a priori* adaptés au problème de segmentation. Ainsi, dans ce travail nous proposons d'unifier les avantages de ces deux types de modèles markoviens pour proposer un système hybride. Nous montrerons que ce système hybride permet d'intégrer des connaissances *a priori* tout en étant robuste à différentes sources de variabilités. Ce système hybride est également générique, nous l'avons testé avec différents types de gestes. L'élaboration de ce système hybride constitue la contribution principale de ce travail de thèse.

Comme nous l'avons mentionné dans la section précédente, il existe des modèles markoviens dédiés au suivi des objets mobiles comme les filtres particulaires [49]. Nous avons appliqué ces modèles markoviens pour caractériser les gestes en suivant la main ayant un mouvement dominant. Notre contribution principale sur ce thème est l'élaboration d'une composante d'amélioration du suivi des filtres particulaires basée sur l'exploitation de l'information de mouvement. L'in-

formation primaire du mouvement est représentée par les vitesses des pixels, qui peuvent être extraites avec une méthode connue pour sa robustesse à la variation de la luminosité et aux déformations de la cible [15], et que l'on désigne par les méthodes des flots optiques [19]. Nous proposons également une méthode automatique de génération du vocabulaire de la main nécessaire pour le suivi avec filtres particulaires.

Néanmoins, cette méthode de suivi a montré ses limites dans le cadre de la caractérisation des gestes. Nous proposons ainsi une alternative sous forme d'une méthode originale de caractérisation globale du mouvement dans les gestes. Cette méthode décrit la localisation, la vélocité et l'orientation du mouvement. Elle est basée sur l'exploitation de l'information de vitesse fournie par les flots optiques.

Pour évaluer les approches proposées, nous avons exploité deux bases de vidéos de gestes filmées avec une caméra fixe. La première base de données, appelée « `SignStream` » [97], est une base spécialisée dans la langue des signes américaine. Nous avons exploité cette base pour évaluer notre approche de suivi de gestes. La deuxième base de données est une base constituée d'un ensemble de sous-bases de gestes appartenant à plusieurs thèmes de communication visuo-gestuelle. Cet ensemble de bases de données est fournies dans le cadre de la compétition « *Gesture Challenge 1-2* » proposée par l'organisation `ChaLearn` en 2011-2012 [54] dont le sujet est la reconnaissance de gestes avec un seul exemple d'apprentissage, tâche que l'on désigne également « *one shot-learning* » [54, 153]. Nous avons exploité ces bases pour évaluer notre système complet de reconnaissance de gestes. Nous allons montrer par la suite que le manque de données d'apprentissage est un autre problème que les modèles markoviens sont capables de résoudre.

1.5 Organisation du manuscrit

Le manuscrit se compose de deux grandes parties. La première partie est dédiée au suivi des gestes. Nous avons étudié le suivi des gestes dans le cadre de la caractérisation des gestes qui est une étape nécessaire pour la reconnaissance des gestes. Cette première partie se compose de deux chapitres. Le premier chapitre présente l'état de l'art des modèles markoviens dédiés au suivi, et en particulier les filtres particulaires et leurs applications. Nous donnons à la fin du chapitre un aperçu sur un ensemble de méthodes de caractérisation des objets nécessaires pour le suivi. Le deuxième chapitre développe notre application des filtres particulaires au suivi de la main dominante dans la langue des signes et les contributions apportées.

La deuxième partie de ce manuscrit est dédiée à la reconnaissance des gestes. Elle se compose de deux chapitres également. Le premier chapitre présente l'état de l'art des modèles markoviens dédiés à la reconnaissance des données séquentielles. Nous expliquons le principe des modèles de Markov cachés et le principe

des champs aléatoires conditionnels. Nous donnons un aperçu de plusieurs variantes associées et nous détaillons certaines applications de ces modèles pour la reconnaissance des gestes. Le deuxième chapitre présente notre application de ces modèles markoviens à la reconnaissance des gestes et les contributions apportées.

Première partie

Suivi des gestes

Chapitre 2

État de l'art : Modèles Markoviens de suivi

Table des matières

2.1 Introduction générale sur les méthodes de suivi

La reconnaissance des gestes nécessite une étape de caractérisation. La caractérisation des gestes est réalisable avec les techniques de suivi des gestes qui sont en général les techniques de suivi des objets. En effet, un geste est généré principalement par le mouvement et la forme des mains. Les trajectoires des mains peuvent ainsi suffire à décrire le geste. Pour extraire la trajectoire des mains, il faut recourir aux techniques de suivi des objets mobiles.

Selon Klein [71], il existe trois catégories de méthodes de suivi : les méthodes d'optimisation d'une grandeur scalaire, les méthodes d'appariement de détection et les méthodes probabilistes.

Les méthodes d'optimisation d'une grandeur scalaire, telles que les méthodes de « template matching » [58], repose sur la minimisation ou la maximisation d'une fonction à valeur dans $\mathbb{R}$ qui présente la correspondance entre un modèle de l'objet et une observation extraite de l'image. Cette fonction peut être représentée par une distance [32], une corrélation [137], une énergie [40], une erreur [50] ou un coût [122].

Les méthodes d'appariement de détection [67, 124, 24, 34, 47, 117], comme la méthode MHT (« *Multiple Hypothesis Tracking* ») [112], reposent sur le principe de correspondance entre des régions détectées de l'objet dans l'image à l'instant $t-1$ et des régions candidates à l'image t.

Les méthodes probabilistes de suivi cherchent la position la plus probable de l'objet suivi dans l'image actuelle à travers le calcul d'une probabilité conditionnelle dépendant de l'historique du déplacement de l'objet et des observations extraites. Dans ses travaux de thèse [71], Klein classifie les méthodes probabilistes de suivi en deux catégories : les méthodes probabilistes à base de filtrage optimal [7, 6] et les méthodes probabilistes à base d'apprentissage de modèle [37, 64]. Les méthodes probabilistes basées sur le filtrage optimal sont représentées par deux méthodes ; les filtres de Kalman [66, 98] et les filtres particulaires [49, 70, 60]. Les méthodes de filtrage optimal calculent une estimation de la position de l'objet suivi avec une modélisation gaussienne pour les filtres de Kalman et un échantillonnage de type Monte Carlo pour les filtres particulaires.

Dans le cadre du suivi des gestes, un geste est souvent caractérisé par des mouvements rapides et irréguliers des mains. Ainsi, le suivi des mains dans un geste nécessite une méthode non-déterministe fournissant une modélisation flexible du mouvement. Étant données que les méthodes d'optimisation d'une grandeur scalaire sont des méthodes déterministes, elles ne sont donc pas adaptées pour suivre les mouvements rapides de la cible. D'autre part, les méthodes d'appariement de détection requièrent en général une étape de segmentation robuste qui peut nécessiter, selon la méthode de segmentation, une étape d'apprentissage. Il est connu que l'apprentissage augmente la complexité de la méthode et nécessite une base d'apprentissage étiquetée dont la construction est en générale coûteuse,

particulièrement dans le cas d'étiquetage de positions d'objets dans des vidéos. Ainsi, les méthodes probabilistes semblent mieux adaptées au suivi des gestes. Ces méthodes sont non-déterministes et ne nécessitent pas une étape d'apprentissage. Comme nous nous intéressons aux modèles markoviens, nous orientons notre étude vers les méthodes probabilistes de filtrage optimal qui intègrent le principe Markovien. Comme l'estimation de la position de la cible avec les filtres particulaires est réalisée avec un processus d'échantillonnage, la modélisation du déplacement de la cible est davantage flexible avec les filtres particulaires. Les filtres de Kalman restreignent cette approximation au modèle gaussien. Ainsi, les filtres particulaires semblent les plus adaptés au suivi des gestes.

Nous présenterons dans le chapitre 3, notre application des filtres particulaires pour le suivi de la main dominante dans la langue des signes. Dans ce chapitre 2, nous présentons en premier lieu le principe des méthodes de filtrage optimal (section 2.2). Ensuite, nous détaillons le principe des filtres particulaires, leurs varaintes et certaines de leurs applications dans le cadre du suivi des gestes (section 2.3). Nous donnons un aperçu sur le principe des filtres de Kalman dans l'annexe A (section A.1). D'autre part, le processus des filtres particulaires nécessite une composante de caractérisation de la cible. Nous donnons ainsi, dans la section 2.5, un aperçu sur l'état de l'art de la caractérisation des objets. Nous terminerons ce chapitre par la présentation des difficultés caractérisant l'évaluation des performances des systèmes de suivi (section 2.6).

2.2 Principe des méthodes probabilistes de suivi par filtrage optimal

Les méthodes probabilistes de suivi par filtrage optimal cherchent la position de l'objet dans une image I_t sachant tous les états précédents $Y_{0:t-1}$ et toutes les observations disponibles depuis le départ $X_{0:t}$. Cette position est représentée par un vecteur d'état Y_t. Ces méthodes probabilistes cherchent donc à maximiser $P(Y_t|X_{0:t}, Y_{0:t-1})$. En particulier, le but du processus de filtrage optimal est de calculer la densité de filtrage $p(Y_{0:t}|X_{0:t})$. Le filtrage de Bayes permet d'obtenir directement une densité $p(Y_t|X_{0:t})$ proche de la densité de filtrage sous certaines hypothèses de dépendances représentées sur la figure 2.1 :

- la densité à l'instant 0 est disponible *a priori* : $p(Y_0)$.
- l'état présent Y_t dépend uniquement de l'état qui le précède ce qui correspond à l'hypothèse markovienne d'ordre 1 : $p(Y_t|Y_{0:t-1}, X_{0:t}) = p(Y_t|Y_{t-1})$.
- l'observation courante, X_t ne dépend que de l'état présent Y_t ce qui conduit à : $p(X_t|Y_{0:t}, X_{0:t-1}) = p(X_t|Y_t)$.

Ainsi le principe d'estimation de la suite des positions ou états d'un objet suivi selon le filtrage optimal se compose classiquement de deux étapes principales : la `prédiction` et la `mise à jour`.

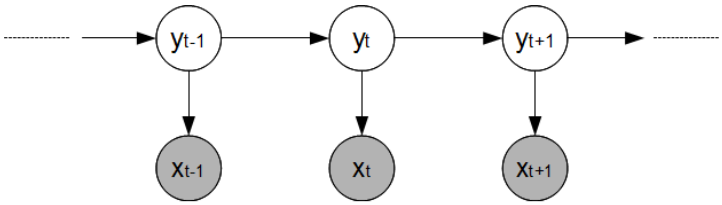


FIGURE 2.1 – Graphe de dépendance du filtrage optimal

La prédiction est réalisée en appliquant la formule de Chapman-Kolmogorov :

$$\begin{aligned} p(Y_t|X_{0:t-1}) &= \int p(Y_t|Y_{t-1}, X_{0:t-1})p(Y_{t-1}|X_{0:t-1})dY_{t-1} \\ &= \int p(Y_t|Y_{t-1})p(Y_{t-1}|X_{0:t-1})dY_{t-1} \end{aligned} \tag{2.1}$$

où $p(Y_{t-1}|X_{0:t-1})$ est déduite de l'itération précédente et $p(Y_t|Y_{t-1})$ représente la probabilité de transition supposée connue.

La mise à jour est réalisée en utilisant la formule de Bayes :

$$\begin{aligned} p(Y_t|X_{0:t}) &= \frac{p(X_t|Y_t)p(Y_t|X_{0:t-1})}{p(X_t|X_{0:t-1})} \\ &= \frac{p(X_t|Y_t)p(Y_t|X_{0:t-1})}{\int p(X_t|Y_t)p(Y_t|X_{0:t-1})} \end{aligned} \tag{2.2}$$

Le dénominateur de l'équation 2.2 est considéré comme une constante de normalisation, $p(X_t|Y_t)$ représente une vraisemblance définie par le concepteur selon des choix spécifiques aux données de l'application pour définir le vecteur de caractéristiques X_t, il reste à calculer $p(Y_t|X_{0:t-1})$ à travers l'équation 2.1. Néanmoins, le calcul de l'intégrale dans l'équation 2.1 est coûteux dans la pratique, particulièrement en temps réel. Cependant, les filtres de `Kalman` [66] et les filtres `particulaires` (section 2.3) proposent une solution d'approximation de la probabilité $p(Y_t|X_{0:t})$ à travers respectivement une modélisation gaussienne et l'échantillonnage de Monte-Carlo. Le principe de calcul de ces filtres probabilistes se base sur la modélisation suivante :

$$Y_t = f(Y_{t-1}, b_t) \tag{2.3}$$

$$X_t = g(Y_t, c_t) \tag{2.4}$$

où l'équation 2.3 est une équation de transition et l'équation 2.4 est une équation d'observation. f et g sont deux fonctions *a priori* quelconques, b_t et c_t sont deux bruits *a priori* quelconques. Nous montrons dans la section suivante la forme des deux fonctions f et g pour les filtres particulaires. La forme de ces deux fonctions pour les filtres de Kalman est présentée dans l'annexe A (section A.1).

2.3 Filtre particulaire : un modèle de suivi markovien

2.3.1 Théorie

Les filtres particulaires [49, 116] offrent une solution à un problème de filtrage non-linéaire. Ce problème est représenté par les équations 2.3 et 2.4. L'avantage des filtres particulaires est qu'ils ne posent aucune restriction sur les fonctions f et g ou les bruits b_t et c_t. Ils offrent ainsi une flexibilité de modélisation des relations entre états et observations.

L'estimation $\hat{Y}_t$ est obtenue via l'expression de la densité de filtrage $p(Y_{0:t}|X_{0:t})$. Cette densité est estimée avec le principe de Monte Carlo et l'échantillonnage d'importance [71].

2.3.1.1 Principe de Monte-Carlo

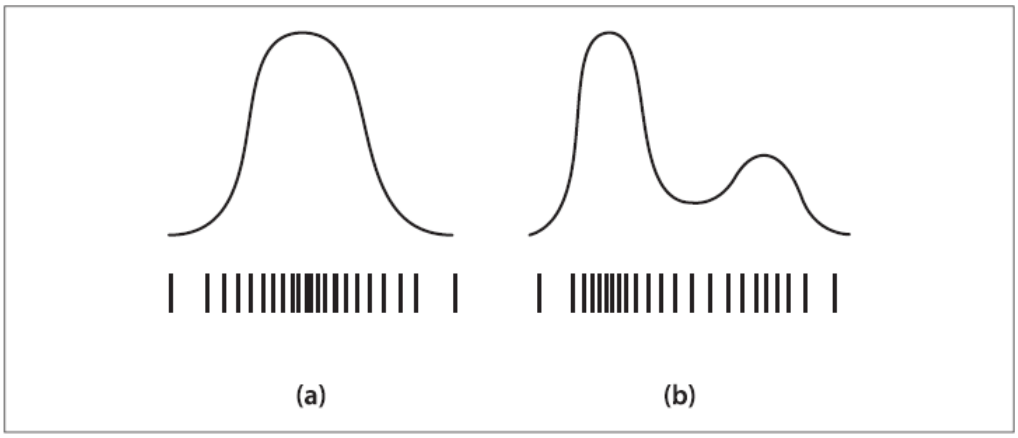


FIGURE 2.2 – Deux exemples de lois de probabilité : (a) une gaussienne qui peut être paramétrée par une moyenne et une variance dans le cas des filtres de Kalman et représentée par une densité d'échantillons dans le cas des filtres particulaires, (b) une loi qui ne peut être représentée que par une densité d'échantillons dans le cas des filtres particulaires [19]

Selon le principe de Motne Carlo [70], il est possible d'estimer une densité de probabilité $p(x)$ représentant la loi suivie par la variable x, définie sur le domaine D_x, en utilisant un ensemble d'échantillons $u^{(i)}$ tirés selon la densité $p(x)$ comme le montre l'équation 2.5. Ces échantillons sont appelées `particules` dans le cas des filtres particulaires. L'estimation d'une densité de probabilité avec ces particules est l'équivalent de la modélisation gaussienne proposée par les filtres de Kalman. La figure 2.2 montre l'avantage de cet échantillonnage qui permet de modéliser une distribution multimodale contrairement au modèle gaussien du filtre de Kalman.

$$\hat{p}(x \in dx) = \frac{1}{N} \sum_{i=1}^{N} \delta_{u^{(i)}}(x) \tag{2.5}$$

où $\delta_{u^{(i)}}$ est la distribution de Dirac centrée en $u^{(i)}$ et dx le différentiel de x. Cette estimation est un cas particulier de la loi forte des grands nombres : $\forall f(.)p(.)$ intégrable,

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} f(u^{(i)}) \xrightarrow[N \to \infty]{p.s} I = \int_{D_x} f(x)p(x)dx \tag{2.6}$$

où I est l'intégrale de $f(.)p(.)$ sur D_x. Dans le cas où $p(x)$ n'est pas connue, `l'échantillonnage d'importance` fourni une solution d'approximation à travers une `densité d'importance` que nous présentons dans la sous-section suivante.

2.3.1.2 Échantillonnage d'importance

Soit une distribution q connue et liée à p de la manière suivante : si $p > 0$ alors $q > 0$, on peut ainsi écrire :

$$I = \int_{D_x} f(x) \frac{p(x)}{q(x)} q(x)dx \tag{2.7}$$

Soit $g(x) = f(x)\frac{p(x)}{q(x)}$. Soient N échantillons $e^{(i)}$ tirés selon q, l'estimateur de Monte Carlo peut alors s'écrire :

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} g(e^{(i)}) \tag{2.8}$$

d'où

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} f(e^{(i)}) \frac{p(e^{(i)})}{q(e^{(i)})} \tag{2.9}$$

Soit $w^{(i)} = \frac{p(e^{(i)})}{q(e^{(i)})}$, les $w^{(i)}$ représentent les `poids d'importance`, plus $w^{(i)}$ est grand, plus $Y^{(i)}$ « suit » la loi p, et donc plus elle est « importante » [71]. Ainsi, q est appelée `densité d'importance`. Avec la normalisation l'estimation devient :

$$\hat{I} = \frac{\frac{1}{N} \sum_{i=1}^{N} f(e^{(i)}) w^{(i)}}{\frac{1}{N} \sum_{i=1}^{N} w^{(i)}} = \sum_{i=1}^{N} f(e^{(i)}) \tilde{w}^{(i)} \tag{2.10}$$

où $\tilde{w} = \frac{w^{(i)}}{\sum_{i=1}^{N} w^{(i)}}$. Ainsi, selon le principe de Monte Carlo et l'échantillonnage d'importance, et sans la normalisation des poids, la densité de filtrage peut s'écrire :

$$\hat{p}(Y_{0:t}|X_{0:t}) = \sum_{i=1}^{N} \delta_{Y^{(i)}}(Y)\tilde{w}_t^{(i)} \tag{2.11}$$

où

$$\tilde{w}_t^{(i)} \propto \frac{p(Y_{0:t}^{(i)}|X_{0:t})}{q(Y_{0:t}^{(i)}|X_{0:t})} \tag{2.12}$$

2.3.1.3 Estimation récursive des poids

L'échantillonnage d'importance est réalisé de manière séquentielle à travers une relation de récurrence entre les poids $w_t^{(i)}$ et $w_{t-1}^{(i)}$. L'équation 2.15 représente cette relation de récurrence.

En effet, d'après la règle de Bayes, $p(Y_{0:t}|X_{0:t})$ peut s'écrire comme suit :

$$p(Y_{0:t}|X_{0:t}) = \frac{p(X_t|Y_{0:t}, X_{0:t-1})p(Y_t|Y_{0:t-1}, X_{0:t-1})p(Y_{0:t-1}|X_{0:t-1})}{p(X_t|X_{0:t-1})} \tag{2.13}$$

Avec les hypothèses markoviennes représentées par la figure 2.1, l'équation 2.13 devient :

$$\begin{aligned} p(Y_{0:t}|X_{0:t}) &= \frac{p(X_t|Y_t)p(Y_t|Y_{t-1})p(Y_{0:t-1}|X_{0:t-1})}{p(X_t|X_{0:t-1})} \\ &= \alpha\ p(X_t|Y_t)p(Y_t|Y_{t-1})p(Y_{0:t-1}|X_{0:t-1}) \end{aligned} \tag{2.14}$$

où $\alpha = \frac{1}{p(X_t/X_{0:t-1})} = \frac{1}{p(X_t)}$ est une constante (le passage de $p(X_t/X_{0:t-1})$ à $p(X_t)$ est justifié par l'hypothèse d'indépendance des observations représentée dans la figure 2.1).

q est choisi afin de respecter :

$$q(Y_{0:t}^{(i)}|X_{0:t}) = q(Y_t^{(i)}|Y_{0:t-1}^{(i)}, X_{0:t-1})q(Y_{0:t-1}^{(i)}|X_{0:t-1})$$

Ainsi, la formule des poids 2.12 devient :

$$\begin{aligned} w_t^{(i)} &= \frac{\alpha\ p(X_t|Y_t^{(i)})p(Y_t^{(i)}|Y_{t-1}^{(i)})p(Y_{0:t-1}^{(i)}|X_{0:t-1})}{q(Y_t^{(i)}|Y_{0:t-1}^{(i)}, X_{0:t})q(Y_{0:t-1}^{(i)}|X_{0:t-1})} \\ &= \frac{\alpha\ p(X_t|Y_t^{(i)})p(Y_t^{(i)}|Y_{t-1}^{(i)})}{q(Y_t^{(i)}|Y_{0:t-1}^{(i)}, X_{0:t})} w_{t-1}^{(i)} \end{aligned} \tag{2.15}$$

Cette procédure est appelée dans la littérature par *Sequential Importance Sampling (SIS)*. L'algorithme 1 reprend

les étapes de cette procédure de prédiction particulaire.

Initialisation, $t = 0$:
Échantillonner N particules selon une distribution $q(Y_0)$ et poser $\forall i, w_0^{(i)} = \frac{1}{N}$
pour *t>=1* **faire**
  Échantillonner les particules selon $Y_t^{(i)} \sim q(Y_t^{(i)}/Y_{t-1}^{(i)}, X_t)$
  Mettre à jour les poids selon $w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(X_t/Y_t^{(i)})p(Y_t^{(i)}/Y_{t-1}^{(i)})}{q(Y_t^{(i)}/Y_{t-1}^{(i)}, X_t)}$
  Normaliser les poids selon $\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^{N} w_t^{(i)}}$
  Estimer le nouvel état Y_t en utilisant l'estimation de la distribution $\hat{p}(Y_t|X_{0:t})$, $\hat{Y}_t = \sum_{i=1}^{N} f(Y_t^{(i)})\tilde{w}_t^{(i)}$
fin pour
Fin

Algorithm 1: SIS

2.3.1.4 Problème de dégénérescence

L'algorithme SIS peut conduire à un problème de dégénérescence : après un certain nombre d'itérations, une seule particule peut posséder un poids proche de 1, tandis que les $N-1$ autres auront un poids proche de 0. Ainsi, l'intérêt du filtre qui consiste à explorer l'espace d'états est perdu. Une mesure de la dégénérescence est proposée dans [83], appelée *taille efficace du N-échantillons* dont l'expression est représentée par l'équation 2.16,

$$N_{eff} = \frac{N}{1 + var(w_t^{*(i)})} \tag{2.16}$$

où $w_t^{*(i)} = \frac{p(Y_t^{(i)}/X_{0:t})}{q(Y_t^{(i)}/Y_{t-1}^{(i)}, X_t)}$. Cette mesure de dégénérescence peut être approximée par la formule 2.17. Le risque de dégénérescence est inversement proportionnel à la valeur de $\hat{N}_{eff}$.

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N} (\tilde{w}_t^{(i)})^2} \tag{2.17}$$

Pour remédier à ce problème, deux solutions sont possibles : choisir une fonction d'importance optimale ou bien choisir une fonction d'importance sous-optimale et ajouter une procédure de réechantillonnage à la procédure principale SIS. La première solution est difficile à mettre en oeuvre et a un coût de calcul élevé. Par contre, la deuxième solution est plus efficace [71]. Dans ce cas, un choix possible de la fonction d'importance q est tel que : $q(Y_t^{(i)}|Y_{0:t-1}^{(i)}, X_{0:t}) = p(Y_t^{(i)}|Y_{t-1}^{(i)}, X_t) = p(Y_t^{(i)}|Y_{t-1}^{(i)})$ [71]. Ainsi, la formule des poids 2.15 devient plus simple : $w_t^{(i)} \propto w_{t-1}^{(i)} p(X_t|Y_t^{(i)})$.

L'ajout de la procédure de rééchantillonnage permet de forcer la répartition des poids des particules. Il s'agit d'éliminer les particules de poids faibles et de multiplier les particules de poids forts à chaque itération. Cette procédure peut être ajoutée d'une manière systématique (algorithme 2), l'algorithme est appelé dans ce cas *Sampling Importance Resembling (SIR)* , ou bien conditionnée par un seuil sur la mesure $\hat{N}_{eff}$ ce qui donne l'algorithme générique des filtres particulaires (algorithme 4).

Initialisation, $t = 0$:
Échantillonner N particules selon une distribution $q(Y_0)$ et poser $\forall i, w_0^{(i)} = \frac{1}{N}$
pour *t>=1* **faire**
  Échantillonner les particules selon $Y_t^{(i)} \sim q(Y_t / Y_{t-1}^{(i)}, X_t)$
  Mettre à jour les poids selon $w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(X_t/Y_t^{(i)}) p(Y_t^{(i)}/Y_{t-1}^{(i)})}{q(Y_t^{(i)}/Y_{t-1}^{(i)}, X_t)}$
  Normaliser les poids selon $\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^{N} w_t^{(i)}}$
  Rééchantillonner en tirant N nouvelles particules $Y_t^{\prime(j)}$ parmi les $Y_t^{(i)}$ selon une probabilité $w_t^{(i)}$ et leur affecter les poids $\tilde{w}_t^{(j)} = \frac{1}{N}$ (algorithme 3)
  Estimer l'état : $\hat{Y}_t = \sum_{j=1}^{N} f(Y_t^{(j)}) \tilde{w}_t^{(j)}$
fin pour
Fin

Algorithm 2: SIR

Initialisation, calcul des poids cumulés, $c_1 = 0$:
pour $i = 2$ *à* N **faire**
  $c_i = c_{i-1} + w_t^{(i)}$
fin pour
Prendre un point de départ u_1 selon la loi uniforme $U[0, \frac{1}{N}]$
pour $j = 1$ *à* N **faire**
  $u_j = u_1 + \frac{1}{N(j-1)}$
  tant que $c_i < u_j$ **faire**
    $i = i + 1$
  fin tant que
  Mettre à jour la particule $Y_t^{\prime(j)} = Y_t^{(i)}$ et le poids $w_t^{(j)} = \frac{1}{N}$
fin pour
Fin

Algorithm 3: Le rééchantillonnage multinomial

Plusieurs méthodes de rééchantillonnage sont développées. La méthode la plus simple et la plus directe est le rééchantillonnage multinomial [71]. Elle repose sur la redistribution des échantillons selon la loi multinomiale de paramètres N

et $(w_t^{(i)})_{i=1}^N$. En tirant avec remise N nouvelles particules avec une probabilité $w_t^{(i)}$, on obtient N nouvelles particules indépendantes et identiquement distribuées selon la probabilité $p(Y|X_{0:t})$. On réaffecte les poids de ces variables à la valeur $\frac{1}{N}$. La simulation de cette procédure de rééchantillonnage est donnée par l'algorithme 3.

Initialisation, $t = 0$:
Échantillonner N particules selon une distribution $q(Y_0)$ et poser $\forall i, w_0^{(i)} = \frac{1}{N}$
pour *t>=1* **faire**
  Échantillonner les particules selon $Y_t^{(i)} \sim q(Y_t/Y_{t-1}^{(i)}, X_t)$
  Mettre à jour les poids selon $w_t^{(i)} \propto w_{t-1}^{(i)} \frac{p(X_t/Y_t^{(i)})p(Y_t^{(i)}/Y_{t-1}^{(i)})}{q(Y_t^{(i)}/Y_{t-1}^{(i)}, X_t)}$
  Normaliser les poids selon $\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^N w_t^{(i)}}$
  Calculer $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N (\tilde{w}_t^{(i)})^2}$
  si $\hat{N}_{eff} < seuil$ **alors**
    Rééchantillonner en tirant N nouvelles particules $Y_t'^{(j)}$ parmi les $Y_t^{(i)}$ selon une probabilité $w_t^{(i)}$ et leur affecter les poids $\tilde{w}_t^{(j)} = \frac{1}{N}$ (algorithme 3)
    Estimer l'état : $\hat{Y}_t = \sum_{j=1}^N f(Y_t^{(j)})\tilde{w}_t^{(j)}$
  sinon
    Estimer l'état : $\hat{Y}_t = \sum_{i=1}^N f(Y_t^{(i)})\tilde{w}_t^{(i)}$
  fin si
fin pour
Fin

Algorithm 4: L'algorithme générique des filtres particulaires

2.3.1.5 L'algorithme de ConDensAtion

L'algorithme de `ConDensAtion` (Conditional Density Propagation, algorithme 5) de Isard et al [60] est l'un des premiers filtres particulaires adapté au suivi des objets et le plus cité dans la littérature [71]. Le schéma 2.3 représente graphiquement les étapes de l'algorithme 5 de ConDensAtion. Les cercles représentent les états $Y_t^{(i)}$ et leur taille représente la valeur du poids associé $w_t^{(i)}$. La première étape de cet algorithme consiste à remplacer les particules de poids faible par des particules de poids fort, cela revient à ré-échantillonner les particules selon leurs poids. Ainsi, on retrouve ici le principe de ré-échantillonnage de l'algorithme SIR 2 pris en compte d'une manière implicite et automatique. La deuxième étape consiste à propager les particules selon la probabilité de transition $p(Y_t|Y_{t-1})$. Dans le cas du suivi d'un objet en mouvement, cette probabilité est représentée par une loi de mouvement prédéfinie. La troisième étape consiste à mettre

à jour les poids $w_t^{(i)}$ selon la probabilité d'observation $p(X_t|Y_t)$. Dans le cas du suivi d'un objet mobile, cette probabilité d'observation est également une fonction prédéfinie proportionnelle aux observations mesurées aux sein des particules associées. Dans la version basique de cet algorithme, $w_t^{(i)} = p(X_t|Y_t^{(i)})$. Dans certaines applications de cet algorithme $w_t^{(i)} = w_{t-1}^{(i)} p(X_t|Y_t^{(i)})$. Enfin, un représentant de ces particules $Y_t^{(i)}$ peut être déterminé en calculant la moyenne pondérée des particules pondérées par leur poids $w_t^{(i)}$. Dans le cas de suivi d'un objet mobile, ce représentant donne la position, où en général, l'état prédit de l'objet à l'instant t.

Initialisation à $t = 0$: $c^{(0)} = 0$, échantillonner N particules selon $p(Y_0)$ et poser $\forall l \in [|1, N|], w_0^{(l)} = \frac{1}{N}$
pour $t >= 1$ **faire**
  (1) **Dériver** les particules ayant des poids forts :
  pour $i \in [|1, N|]$ **faire**
    (a) générer un nombre $r \in [0, 1]$ selon une distribution uniforme.
    (b) recherche dichotomique du plus petit $j \in [|1, N|]$ pour lequel la probabilité cumulative $c_{t-1}^{(j)} \geq r$.
    (c) ${Y'^{(i)}}_t = {Y^{(j)}}_{t-1}$.
  fin pour
  (2) **Propager** les particules selon la probabilité $p(Y_t|Y_{t-1})$:
  pour $i \in [|1, N|]$ **faire**
    $Y_t^{(i)} = \mathbf{A}Y_t'^{(i)} + \mathbf{B}v_t^{(i)}$ où A est la matrice de transformation linéaire statique, $v_t^{(i)}$ est un vecteur de variables aléatoires normales standards et BB^T est la covariance du processus de bruit.
  fin pour
  (3) **Mesurer** les observations X_t et **mettre à jour** les poids w_t associés :
  pour $i \in [|1, N|]$ **faire**
    $w_t^{(i)} \propto p(X_t|Y_t^{(i)})$
  fin pour
  Normaliser les poids tel que $\sum_{i=1}^{N} w_t^{(i)} = 1$ selon $\tilde{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{i=1}^{N} w_t^{(i)}}$
  pour $i \in [|1, N|]$ **faire**
    $c_t^{(i)} = c_{t-1}^{(i)} + w_t^{(i)}$
  fin pour
  (4) **Estimer** (si nécessaire) le moment des particules :
  $\hat{Y}_t = \sum_{i=1}^{N} \tilde{w}_t^{(i)} f(Y_t^{(i)})$
fin pour
Fin

Algorithm 5: Algorithme ConDensAtion

Étant donné la simplicité, l'efficacité et l'application étendue de l'algorithme de ConDensAtion pour le suivi des objets, nous l'avons choisi pour suivre la main dominante dans un discours de langues des signes. Nous expliquons dans le prochain chapitre les détails de l'application de cet algorithme et les modifications apportées afin d'améliorer la qualité du suivi.

Dans le cas de suivi par filtrage particulaire avec une représentation des objets par régions, si une seule région de l'objet sort du champ de la caméra, le reste du modèle à base de régions reste valable pour maintenir le suivi sur la partie toujours visible. Le filtre particulaire grâce à l'échantillonnage d'importance permet de tirer

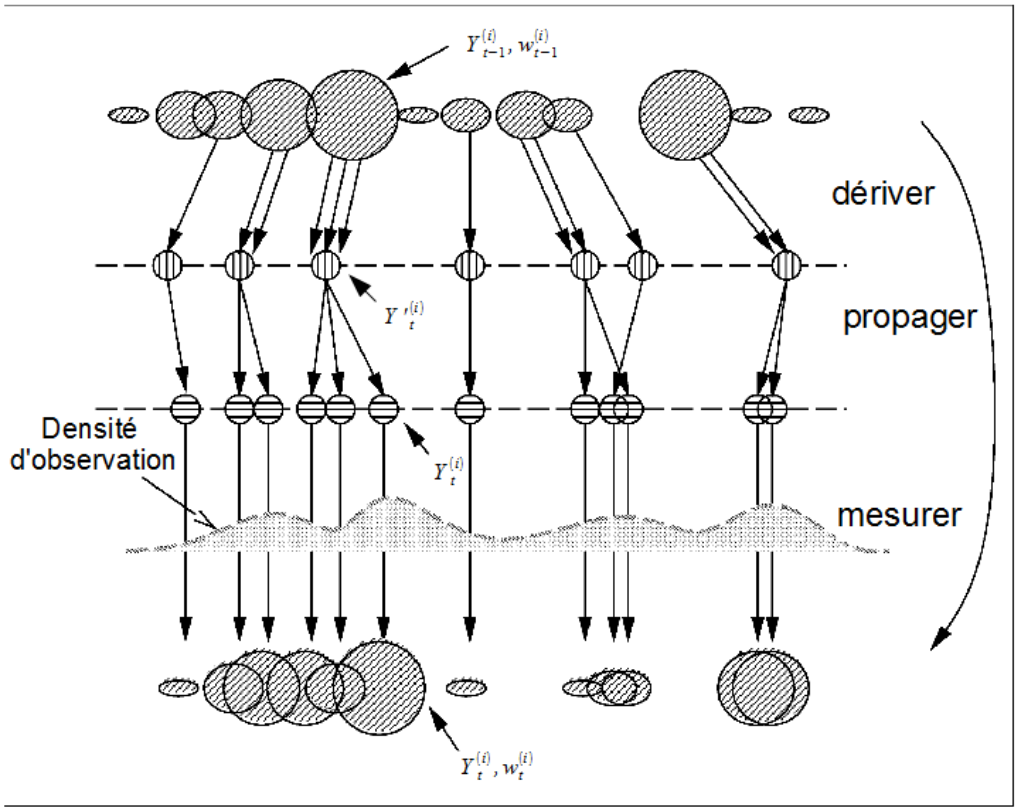


FIGURE 2.3 – Représentation graphique de l'algorithme de Condensation [60]

au hasard des positions éventuelles de l'objet dans une région plus importante (en terme de poids) pour un temps de calcul équivalent. Même si la position tirée au hasard ne permet que de retrouver une partie de l'objet, la convergence du filtre après quelques itérations assure la reprise complète du suivi. Ce principe présente un avantage des filtres particulaires par rapport aux filtres de Kalman. Dans le cas des occultations par exemple, l'objet caché peut changer de vitesse ou de direction. Le filtre de Kalman ne peut pas modéliser ces différentes possibilités. Son modèle gaussien ne permet pas de représenter une distribution multimodale [19].

En résumé, la modélisation de la prédiction de la position de l'objet Y sachant les caractéristiques observés X de la cible est plus flexible avec les filtres particulaires qu'avec les filtres de Kalman grâce au principe de l'échantillonnage de Monte-Carlo.

Dans le but d'optimiser la fonction d'importance (équation 2.9), d'autres variantes des filtres particulaires ont été développées. Nous présentons dans le paragraphe suivant quelques variantes des filtres particulaires pour le suivi des objets. Ces variantes permettent d'améliorer le modèle d'attache aux données qui représente le point faible des filtres particulaires classiques. En d'autres termes, pour

améliorer les performances de suivi des filtres particulaires, il est nécessaire de mieux calculer la pertinence d'une zone à représenter l'objet suivi.

2.3.2 Variantes

Les filtres particulaires sans parfum (Unscented Particle Filtre) [144] modélisent la densité d'importance par une gaussienne dont les paramètres sont déterminés avec un filtre de Kalman pour chaque particule. Le choix de la densité d'importance a une influence directe sur la variance des poids des particules [71] et par la suite sur leur distribution. Néanmoins, avec cette variante, il y a une étape supplémentaire de détermination des paramètres de la densité d'importance.

Les filtres particulaires auxiliaires (FPA) et **les filtres particulaires guidés par les observations** [106, 133] se basent sur le principe de double échantillonnage à travers une variable aléatoire auxiliaire. Le deuxième échantillonnage permet d'orienter les particules vers les zones pertinentes de l'image avant leur propagation. En théorie, cela permet de rapprocher la densité d'importance de sa forme optimale. Néanmoins, les performances de cette variante sont corrélées à la méthode de fusion des observations [71].

Ces approches permettent de contrôler la sélection des particules et par la suite d'améliorer le suivi des filtres particulaires. Dans la suite, nous allons montrer comment les chercheurs ont tenté d'améliorer l'orientation des particules des filtres classiques dans la pratique.

2.4 Applications de suivi avec les filtres particulaires

Les filtres particulaires ont été appliqués avec succès au suivi de différents types d'objets tel que le suivi de véhicules [71], le suivi de sportifs [84, 154], le suivi du visage et de la main [45, 15, 60, 123] et le suivi des doigts de la main [86, 22, 127].

Les filtres particulaires ont été souvent combinés à d'autre mécanismes de gestion des particules pour mieux les orienter vers le chemin réel de l'objet mobile. Shan et al. [123] ont introduit le mécanisme de Mean-Shift [1] dans le processus de prédiction des filtres particulaires pour les orienter davantage vers les zones de concentration de la couleur de l'objet suivi. Le schéma 2.4 illustre cette intégration et montre que, à l'étape d'utilisation du Mean-Shift, les particules sont tirées vers les sommets de la distribution de couleur associée à l'objet suivi. L'effet de ce mécanisme est introduit dans les poids des particules.

1. La méthode du Mean-Shift se base sur une estimation itérative du maxima d'une distribution en utilisant un fonction de pondération, en générale, gaussienne. Ce maxima est décalé à chaque itération vers la moyenne pondérée d'un ensemble d'échantillons appartenant à son voisinage (sous-section A.2.2 de l'annexe A).

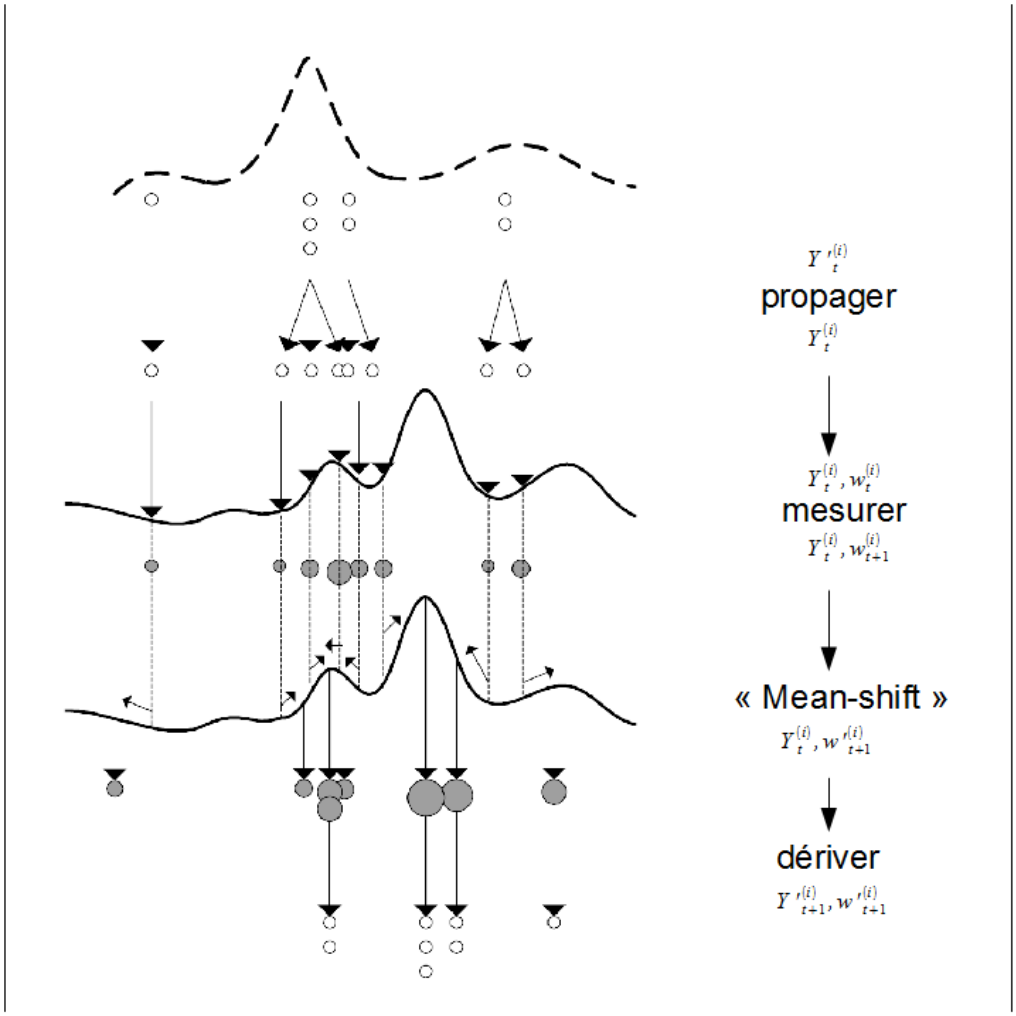


FIGURE 2.4 – Représentation graphique de l'algorithme des filtres particulaires avec intégration de la méthode "Mean-Shift" [123]

Cette idée de combinaison des mécanismes de gestion des particules est adoptée par Gianni et al. [46] également en introduisant le principe du recuit simulé dans l'étape de mise à jour des poids. Le schéma 2.5 illustre cette intégration et la compare à l'étape basique de mise à jour des poids. Le schéma montre que le mécanisme de recuit simulé permet d'explorer davantage l'espace de la densité d'observation et de guider les particules vers des maximas, s'ils existent, plus importants.

Dans les deux exemples précédents, l'amélioration du suivi des particules est introduite dans l'étape de mise à jour de leurs poids. Une autre façon d'améliorer le suivi des particules est de renforcer le modèle de mouvement des particules. Bhandarkar et al. et Yao et al. [15, 154] ont ajouté un terme de vitesse globale calculée avec les flots optiques à l'équation de propagation des particules (équation

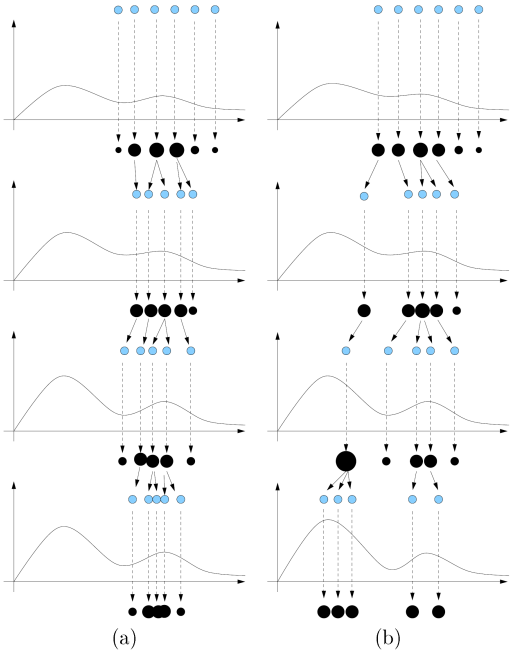


FIGURE 2.5 – Comparaison du déroulement de l'algorithme des filtres particulaires sans (a) et avec (b) l'intégration du principe du recuit-simulé [46]

2.3).

Une deuxième difficulté rencontrée dans la pratique par les filtres particulaires est la détermination de la référence. La référence est un modèle de la cible défini dans un certain espace de caractéristiques dans lequel chaque particule est également écrite. En général, pour chaque particule, la probabilité d'observation est calculée en fonction d'une distance entre les caractéristiques de la particule et les caractéristiques de la référence. Une manière classique pour déterminer la référence consiste à détecter la cible dans l'image initiale de la scène. Yang et al. [152] réalisent cette détection de manière automatique en appliquant les flots optiques, qui est une méthode de caractérisation de mouvement que nous présentons dans la sous-section 2.5.3. Il existe une autre méthode de détermination d'une référence générique qui se base sur l'apprentissage d'un modèle de la cible [105, 60, 46].

En résumé, dans le cadre du suivi avec les filtres particulaires, l'idée principale des contributions dans la littérature est l'intégration d'un mécanisme d'orientation des particules vers les zones les plus pertinentes. Cette intégration est appliquée sur le mécanisme de mise à jour des poids ou le mécanisme de propagation des particules. Néanmoins les propositions d'amélioration mentionnées précédemment présentent un ensembles de difficultés. En effet, l'idée d'attirer les particules vers des maximums de distribution de couleur par exemple risque de déplacer ces particules vers d'autres objets de même caractéristiques que la cible. Comme dans le cas du suivi d'une main, il est possible que les particules soient attirées par la

deuxième main ou par le visage dans la mesure où leurs couleurs sont proches. De même, lorsqu'il y a plusieurs objets mobiles dans une vidéo, il est difficile de lisser le mouvement des particules qui suivent une seule cible avec une seule vitesse globale. Cette vitesse globale reste ambigüe. D'autre part, dans le cas où la cible est un objet déformable, comme le cas de la main, il est difficile de corriger la prédiction du filtre avec une référence déterminée lors d'une étape de détection initiale basée sur une seule pose de la cible. La référence doit être plutôt générique et représentative de plusieurs poses de la cible pour que le filtre puisse suivre ses déformations par la suite. De plus, la méthode de détermination du modèle de la référence avec une méthode d'apprentissage nécessite la construction d'une base d'apprentissage annotée qui est un processus coûteux en général.

Nous avons détaillé depuis le début de ce chapitre le principe des principaux modèles constituant les filtres particulaires : le modèle de mise à jour des poids, appelé également modèle d'observation, le modèle de prédiction, appelé également le modèle de mouvement, et le modèle de référence. Le modèle d'observation n'est pas limité à la probabilité d'observation, il comprend également le modèle de caractérisation de la cible. La comparaison entre les particules et le modèle de référence de la cible est réalisée à travers la comparaison des caractéristiques associées. Nous clôturons ainsi ce chapitre par la présentation d'un aperçu sur l'état de l'art de la caractérisation de la cible.

2.5 Caractérisation de la cible

Il existe trois catégories de descripteurs d'objets exploitables pour la caractérisation de la cible [71] : les descripteurs de couleur et de texture, les descripteurs de forme et les descripteurs de mouvement. Nous donnons dans ce qui suit un aperçu sur chaque catégorie de descripteurs.

2.5.1 Descripteurs de couleur et de texture

Il existe deux descripteurs de couleur fréquemment appliqués qui sont les histogrammes de couleur [155, 129] et la densité de couleur [32, 46].

En ce qui concerne la texture, trois types de descripteurs sont proposés dans la littérature : les méthodes de textons [160, 4], les méthodes statistiques [43, 55] et les méthodes échelle-fréquence [18, 135].

Il existe également des descripteurs hybrides texture-couleur tels que les vecteurs de cohérence couleur [103], les histogrammes chromatiques spatiaux [31] et la transformé de Fourrier quaternionique [94].

Dans le cas de suivi d'objets, les descripteurs de couleurs sont largement utilisés [71]. Les descripteurs de texture ne semblent pas des descripteurs distinctifs pour une main à cause de sa texture uniforme.

La couleur dans une image est représentée selon un espace de couleur spécifique. Il existe quatre espaces principaux de présentation de couleur [71] : l'espace des couleurs primaires RGB (Red Green Blue), l'espace luminance-chrominance YCbCr, l'espace perceptuel HSV (Hue Saturation Value) et l'espace à axes indépendants ACP (Analyse en Composantes Principales).

Il est montré que dans l'espace de couleur `HSV` l'information de couleur est mieux reproduite [157]. Il est également montré que cet espace est robuste aux changements de la luminosité et discriminant entre la couleur de la peau et le fond [125].

2.5.2 Descripteurs de forme

En général, il existe quatre catégories de descripteurs de forme :

- les descripteurs de contour, contenant deux sous catégories ; les descripteurs différentiels [26, 107] et les descripteurs basés sur les modèles déformables [68, 27],
- les descripteurs structurels qui se basent souvent sur une étape de squelettisation [41, 89],
- les descripteurs basés sur les points d'intérêt et les graphes [56, 10],
- les descripteurs statistiques basés sur le calcul des moments [59, 2].

Dans le cas de la caractérisation de forme des objets déformables, les descripteurs basés sur les modèles déformables, qui s'adaptent à la forme variable de l'objet, semblent les plus adéquats. Néanmoins, dans le cas de la caractérisation d'un geste quelconque à travers le suivi d'une main, les détails des contours de la main (les doigts, le palme...) ne sont pas indispensables pour la détermination de la trajectoire de la main. Ainsi, les descripteurs basés sur les modèles déformables qui modélisent les détails de la forme de l'objet semblent complexifier la tâche de caractérisation de la main inutilement. Cependant, une description globale de la forme de la main peut aider le système de suivi à distinguer la main cible des autres objets qui peuvent avoir la même couleur. Notons que la main dans un geste quelconque peut prendre plusieurs poses différentes. Ainsi, il est nécessaire de choisir des descripteurs de forme génériques et invariants aux transformations géométriques. Par la suite, les descripteurs statistiques, comme les moments de Hu [59] et les moments de Zernike [2], semblent les plus adéquats dans ce contexte.

D'après Zhang et al. [159], les moments de Zernike sont les moments les plus performants en matière d'analyse de forme. Ils donnent une description globale de la forme de l'objet. Les moments de Zernike sont des moments complexes et orthogonaux. Leur orthogonalité permet d'éviter la redondance de l'information et de raffiner la description de l'objet. Ces moments sont robustes aux bruits et aux déformations [114]. Ils sont connus pour leur invariance aux rotations. Ils peuvent également être invariants par translation ou par changement d'échelle

en utilisant une méthode de normalisation ou une combinaison d'un ensemble de moments géométriques centraux [30, 69, 17].

Les moments de Hu [59] sont moins robustes que les moments de Zernike en ce qui concerne l'invariance par rotation et la résistance au bruit. Cependant, certaines études [76] ont montré que ces moments peuvent être plus robustes aux translations. En effet, ces moments sont sous forme de combinaisons de moments géométriques centrés et normalisés construits dans le but de vérifier l'invariance par translation. Ils sont également invariant au changement d'échelle et à la symétrie axiale.

2.5.3 Descripteurs de mouvement

Les descripteurs de mouvement sont spécifiques à la caractérisation des objets mobiles. Théoriquement, la caractérisation du modèle de référence pour les filtres particulaires doit être indépendante du modèle de mouvement de la cible. Classiquement, le modèle de référence est décrit dans un repère statique. Néanmoins, les descripteurs de mouvements sont exploitables pour le contrôle de l'orientation des particules et l'amélioration de la qualité du suivi.

Bugeau [25] classifie les descripteurs de mouvement en quatre catégories :

- les méthodes basées sur la différence inter-images qui regroupent les méthodes de détection de mouvement [63, 81] et les méthodes d'estimation d'un champ de déplacement [85, 100],
- les méthodes d'analyse du font de la scène [51, 113],
- les méthodes d'extraction en couches de mouvement [29, 138],
- les méthodes à base de mouvement cohérent [148, 75].

Les flots optiques [85] représentent une méthode d'estimation d'un champ de déplacement, appelé également champ de vitesse. D'après Klein [71], cette méthode est la méthode de description de mouvement la plus connue dans la littérature. Elle présente la base d'un grand nombre de méthodes de caractérisation du mouvement.

Les flots optiques sont des champs de vitesses calculés à partir de deux images successives. Ces vitesses sont les vitesses des pixels qui subissent un changement. Le calcul des flots optiques sert à suivre les objets qui se déplacent dans une scène filmée avec une caméra fixe ou mobile. La figure 2.6 présente un exemple de 4 points repérés sur une image et les vecteurs vitesses associés déterminés avec les flots optiques. Nous pouvons voir également sur la figure 2.6 tout le champ de vitesse calculé avec les flots optiques à partir des deux images. La scène est une scène à contenu fixe filmée par une caméra mobile.

Une méthode classique de calcul des flots optiques est la méthode de Lucas-Kanade [85]. Cette méthode se base sur trois hypothèses : la conservation de la luminosité, la continuité temporelle et la cohérence spatiale. La conservation de la luminosité signifie qu'entre deux images successives la luminosité des pixels ne

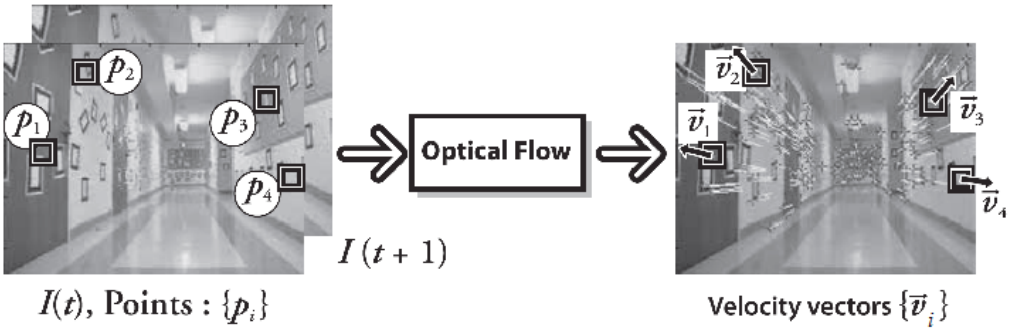


FIGURE 2.6 – Un exemple d'application des flots optiques [19]

change pas au cours de leur déplacement, seule la position change. Néanmoins ce déplacement spatiale, entre deux images successives, reste considérablement petit selon la deuxième hypothèse de continuité temporelle. Enfin, la cohérence spatiale signifie que les pixels d'un même voisinage subissent le même déplacement. En traduisant mathématiquement ces hypothèses, nous pouvons déduire la formule de la vitesse v des flots optiques représentée par l'équation 2.18. La démarche mathématique permettant d'obtenir cette formule de vitesse est détaillée davantage dans [19].

$$\mathrm{v} = (M^T M)^{-1} M^T \mathrm{u} \tag{2.18}$$

où

$$M = \begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_n) & I_y(p_n) \end{bmatrix}, \qquad \mathrm{v} = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad \text{et} \quad \mathrm{u} = \begin{bmatrix} -I_t(p_1) \\ -I_t(p_2) \\ \vdots \\ -I_t(p_n) \end{bmatrix}$$

$p_1, p_2, \ldots, p_n$ sont les pixels inclus dans la fenêtre de calcul, et $I_x(p_i), I_y(p_i), I_t(p_i)$ sont les dérivées partielles de l'image I selon les variables d'espace x, y et de temps t, évaluées au point p_i et au temps courant [2].

Les flots optiques sont connus pour leur robustesse vis à vis des variations de luminosité et des déformations de l'objet suivi [15]. Ils représentent ainsi un descripteur adéquat à la caractérisation d'une main mobile avec une forme variable.

2. http ://fr.wikipedia.org/wiki/M%C3%A9thode_de_Lucas%E2%80%93Kanade

2.6 Difficultés d'évaluation des performances des méthodes de suivi

Nous avons remarqué que dans la littérature, il n'existe pas de test standardisé pour l'évaluation des méthodes de suivi, en particulier, les filtres particulaires. La difficulté d'annotation des bases de données de suivi, qui nécessite en général de localiser manuellement la cible dans chaque trame vidéo, est une des raisons. Souvent les chercheurs évaluent visuellement les algorithmes de suivi ; ils présentent un ensemble de trames d'images avec une représentation visuelle du résultat de localisation de la cible avec l'algorithme de suivi. Klein le confirme dans ses travaux de thèse [71] en expliquant cette difficulté d'évaluation par :

- *la subjectivité de la qualité des résultats : difficile de définir un « bon suivi », cette notion est laissée à l'appréciation de l'utilisateur humain.*
- *l'hétérogénéité des algorithmes : si deux algorithmes utilisent une représentation différente de l'objet, il est difficile de comparer leurs performances. Il en va de même pour les séquences traitées.*

Bashir et al. ont récemment proposé dans [11] une mesure d'évaluation de la qualité de suivi qu'ils ont appelée le *« taux de suivi »* et noté par r_t. Ce taux de suivi permet de déterminer le pourcentage de superposition entre la région estimée de l'objet et la région réelle de l'objet. Il est donné par l'équation 3.10. Cette mesure rappelle l'indice de Jaccard [61].

$$r_t = \frac{2 \times S(A_t \cap B_t)}{S(A_t) + S(B_t)} \tag{2.19}$$

où $S(.)$ est une fonction retournant la surface, A_t représente la région de la cible estimée par l'algorithme de suivi, B_t représente la région de la cible définie dans la vérité terrain. Ce taux de suivi est compris entre 0 et 1. Le tableau 2.1 représente l'interprétation des valeurs de ce taux de suivi lorsqu'il est moyenné sur toute la vidéo selon les auteurs.

$0 \leq \bar{r_t} < 0.4$	$0.4 \leq \bar{r_t} < 0.7$	$0.7 \leq \bar{r_t} < 0.9$	$0.9 \leq \bar{r_t} \leq 1$
suivi médiocre	suivi partiel	suivi globalement satisfaisant	suivi excellent

TABLE 2.1 – Interprétation des valeurs du taux de suivi moyen [71]

Bashir et al. ont également proposé d'évaluer le suivi avec le calcul d'une erreur en utilisant la distance euclidienne entre le centre de A_t et le centre de B_t. Cette erreur est définie si et seulement si $A_t \cap B_t \neq \varnothing$.

Conclusion

Nous avons introduit dans ce chapitre les différentes de méthodes de suivis d'objet et nous avons détaillé, en particulier, le principe des filtres particulaires.

Nous avons également présenté les principales varaintes de cette méthodes et un ensemble de ses applications pour le suivi d'une main. D'après les tentatives d'amélioration apportées à cette méthode de suivi proposées dans la littérature, nous avons déduit que le point faible des filtres particulaires est le modèle d'attache aux données. Il s'avère nécessaire d'introduire un mécanisme de contrôle et d'orientation des particules vers les zones les plus pertinentes afin d'améliorer le suivi, particulièrement dans le cas de mouvements rapides et irréguliers de la cible. Dans ce cadre, nous proposons d'introduire dans le processus de pondération des particules un mécanisme de pénalisation des particules basé sur des informations de vitesse extraites avec les flots optiques. Nous donnons les détails de cette contribution dans le prochain chapitre. Nous montrerons également comment nous avons utilisé la même source d'information de vitesse pour contrôler le modèle de mouvement des particules. Nous proposons également une méthode de détermination du modèle de référence adaptée à une cible déformable avec une pose variable sans avoir recours à une étape d'apprentissage.

Dans ce chapitre, nous avons également donné un aperçu sur l'état de l'art de la caractérisation d'objet, étape nécessaire pour le processus du suivi avec les filtres particulaires. Comme dans notre application de suivi avec les filtres particulaires, la cible est une main dominante représentant un objet déformable avec une pose variable, il est nécessaire de choisir des descripteurs invariants aux déformations. Nous adoptons ainsi les descripteurs de couleur ; les histogrammes de couleur et la densité de couleur à travers l'application de la méthode Camshift, qui représente une extension de la méthode Mean-shift (section A.2 de l'annexe A), et des descripteurs de forme statistiques ; les moments de Zernike et les moments de Hu.

Enfin, nous avons présenté les difficultés d'évaluation des performances des systèmes de suivi et les mesures d'évaluations proposées par Bashir et al. dans [11]. Nous adoptons des métriques d'évaluation proches de ces mesures, principalement la mesure de taux de suivi qui nous permettra de classifier les résultats de suivi de notre système.

Nous détaillerons dans le prochain chapitre notre application de ce modèle markovien et les contributions apportées pour répondre au problème d'orientation des particules vers les zones les plus pertinentes et d'amélioration du mécanisme de suivi. Le but de notre application de suivi est la caractérisation des gestes à travers le suivi d'une ou plusieurs composantes corporelles générant les gestes.

Chapitre 3

Application des filtres particulaires pour la caractérisation des gestes : amélioration du suivi avec les flots optiques

Table des matières

Introduction

Le but de notre travail de suivi n'est pas le suivi en lui-même mais la caractérisation des gestes. Le suivi d'une ou de plusieurs composantes corporelles, comme les mains, caractérise le geste effectué. Dans le cadre d'étude des gestes, le thème classique le plus étudié est la langue des signes [128, 36, 141, 142, 21, 96]. Par la suite, les chercheurs dans ce domaine ont tenté de construire des bases de données accessibles comme la base de la langue des signes américaine « SignStream » [97]. Ainsi, nous avons choisi d'appliquer les filtres particulaires pour le suivi de la main dominante dans la langue des signes principalement du fait de la disponibilité de cette ressource au moment où ce travail a débuté. Nous précisons que nous n'avons pas intégré des connaissances linguistiques spécifiques à la langue des signes. Nous avons orienté notre étude vers la résolution du problème du suivi des mouvements rapides et irréguliers caractérisant la langue des signes, tout en conservant dans notre démarche une approche générique.

Dans ce chapitre nous proposons un modèle markovien pour le suivi de la main dominante dans la langue des signes. Ce modèle est une amélioration de la variante classique des filtres particulaires à l'aide d'une méthode de pénalisation originale basée sur les flots optiques.

Comme nous l'avons déduit dans le chapitre précédent, le problème des filtres particulaires est la possibilité de divergence des particules, notamment dans le cas de mouvements rapides et irréguliers. Ce type de mouvements est fréquent dans les gestes de la langue des signes.

En effet, la main dominante dans la langue des signes est le composant corporel qui réalise principalement le signe. C'est l'objet de la scène qui a en général le mouvement le plus intense et le plus fréquent. Cette caractéristique de dominance va être introduite avec d'autres informations de mouvement dans le modèle de mouvement des particules afin de le contrôler.

Afin de renforcer le modèle d'attache aux données des filtres particulaires et de contrôler davantage l'orientation des particules, nous avons introduit une étape de pénalisation des particules dans le modèle d'observation, ajoutée spécifiquement à l'étape de mise à jour des poids des particules. Cette étape de pénalisation repose sur des informations de vitesses extraites avec les flots optiques [85]. L'idée d'intégrer un mécanisme de contrôle des particules dans l'étape de mise à jour des poids existe dans la littérature comme expliqué dans la sous-section 2.4 du chapitre précédent. L'originalité de notre méthode dans ce cadre est l'application

des flots optiques pour la pénalisation des particules. La deuxième originalité est le contrôle du modèle de mouvement des particules à travers un auto-lissage du mouvement pondéré par des coefficients calculés avec les flots optiques.

Une troisième contribution proposée dans ce chapitre est l'adaptation du modèle de référence au cas du suivi d'un objet déformable sans avoir recours à une étape d'apprentissage. Le modèle de référence est le représentant de la cible nécessaire pour évaluer la similarité entre le contenu de la particule et la cible. La figure 3.1 résume les étapes de suivi avec les filtres particulaires et les contributions que nous avons apportées (« auto-lissage », « pénalisation avec les flots optiques » et « vocabulaire de référence »).

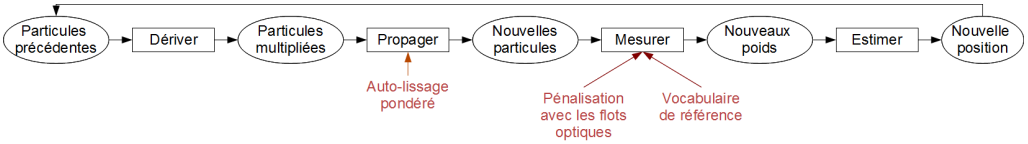


FIGURE 3.1 – Les étapes des filtres particulaires avec les contributions réalisées

Ce chapitre est organisé de façon à présenter les trois composantes principales des filtres particulaires conçues selon notre application : le modèle de référence (section 3.1), le modèle d'observation (section 3.2) et le modèle de mouvement(section 3.3). L'algorithme des filtres particulaires adopté dans notre cas est l'algorithme de ConDensAtion de Isard et al. [60]. Nous présentons ensuite dans la section 3.4 les expérimentations réalisées et dans la section 3.5 les principaux résultats obtenus qui montrent l'apport de nos contributions. Concernant l'étape d'évaluation de notre système de suivi, bien qu'il existe des bases de la langue des signes accessibles comme la base SignStream, l'annotation de la position exacte de la main dominante sur les trames des vidéos de SignStream n'est pas fournie, nous avons donc été amenés à annoter manuellement un ensemble de trames de cette base, afin de réaliser les évaluations.

3.1 Détermination du modèle de référence

Le modèle de référence est un modèle de la cible indispensable à la correction du parcours des particules qui suivent cette cible. Il constitue un élément essentiel du processus de pondération des particules. Ce processus de pondération des particules permet de sélectionner à chaque itération de l'algorithme les particules les plus similaires au modèle de la référence afin de les propager.

Comme nous l'avons présenté dans le chapitre précédent (section 2.4), il existe deux méthodes principales pour déterminer le modèle de référence : une détection de la cible dans l'image initiale de la vidéo sujet du suivi, ou un apprentissage préalable d'un modèle de la cible. La première méthode est difficile à appliquer pour une cible déformable avec un changement continu de sa forme durant la

vidéo, comme c'est le cas d'une main dans un discours de langue des signes. La deuxième méthode nécessite un processus d'apprentissage avec une base étiquetée ce qui est coûteux à construire en général.

Ainsi, afin de décrire efficacement la variabilité de l'objet suivi sans avoir recours à l'apprentissage, nous proposons une méthode de détermination d'un modèle de référence basée sur une construction automatique d'un vocabulaire de la cible à partir duquel une référence générique est déterminée. Ce vocabulaire est constitué d'un ensemble d'imagettes représentant différentes poses de la main collectées d'une manière automatique, à intervalles de temps réguliers, dans la vidéo où le suivi va être appliqué. Notre méthode ne nécessite pas d'étape d'apprentissage et permet d'éviter par la suite l'annotation des données. En général, l'annotation des données n'est pas requise pour les filtres particulaires puisque par définition ils ne nécessitent pas une étape d'apprentissage.

Pour construire automatiquement ce vocabulaire du modèle de référence, nous avons conçu un algorithme de détection de la main basé sur l'extraction d'un modèle de la couleur de la peau du signeur des gestes dans la vidéo. La couleur de la peau est un indicateur robuste aux déformations de la main. Cependant, cet algorithme de détection ne peut pas être appliqué pour un suivi complet de la main puisque son efficacité diminue dans certaines situations ambigües comme dans le cas où les deux mains se superposent ou dans le cas où la main se rapproche du visage. Néanmoins, cet algorithme peut être exploité pour collecter quelques poses de la main tout en évitant ces situations ambigües. La construction du vocabulaire est une étape préalable au suivi. Autrement dit la vidéo est parcourue une première fois pour la collecte du vocabulaire et une deuxième fois pour la réalisation du suivi. Le premier parcours n'est pas complet, seules certaines poses de la main sont choisies d'une manière périodique sur la vidéo. La figure 3.2 représente un échantillon extrait d'un vocabulaire de la main dominante collecté d'une vidéo de discours en langue des signes.

FIGURE 3.2 – Un échantillon d'un vocabulaire de la référence collecté d'une manière automatique d'une vidéo en langue des signes

Cet algorithme contient principalement trois étapes. Nous commençons par extraire un modèle de la couleur de la peau en appliquant la méthode de détection de visage de Viola et Jones [139] sur la trame initiale de la vidéo. Ensuite, sur une trame, nous déterminons l'ensemble des zones de la couleur de peau en utilisant la méthode de rétro-projection, des opérations morphologiques et une méthode de collecte des contours [19]. Puis, avec des connaissances *a priori* et avec des

opérations géométriques simples basées sur la taille des zones détectées et leur localisation, nous déterminons la zone la plus probable qui peut correspondre à la main dominante. Cette méthode de détection de la main est utilisée également pour détecter la main dans la trame initiale de la vidéo afin d'initialiser le filtre particulaire. Nous rappelons que cet algorithme de détection ne peut pas être appliqué pour un suivi complet de la main puisque son efficacité diminue dans certaines situations ambigües comme dans le cas où les deux mains se superposent ou dans le cas où la main se rapproche du visage.

Après l'étape de collecte du vocabulaire de la référence vient l'étape de caractérisation de la référence. Le calcul des poids se base sur la comparaison des caractéristiques de la référence et les caractéristiques des particules pour évaluer la similarité des particules à la référence. Cette comparaison se fait séparément pour chaque particule. Dans notre cas, la référence est représentée par plusieurs instances de la cible contenues dans le vocabulaire. Ainsi, deux cas se présentent : soit comparer la particule avec chaque imagette du vocabulaire et choisir la distance minimale, soit fusionner les caractéristiques de toutes les imagettes et générer un seul représentant du vocabulaire puis le comparer avec les particules. Notons que le vocabulaire de la référence contient un nombre raisonnable de poses de la référence par rapport au nombre de différentes poses qui peuvent exister dans la vidéo. Ainsi, d'autres poses contenues dans la vidéo restent non représentées. Ainsi, une comparaison restreinte aux poses du vocabulaire semble insuffisante. En revanche, si nous calculons un représentant du vocabulaire qui fusionne les caractéristiques des poses contenues dans ce vocabulaire, il sera générique et sera capable de représenter implicitement plusieurs allures de la cible même non existantes dans le vocabulaire. Un représentant simple qui peut fusionner ces allures est l'opérateur « moyenne ». En effet, nous avons testé le cas de comparaison élémentaire, pose par pose, et le cas de comparaison avec un représentant moyen et nous avons constaté que l'opérateur moyen donne de meilleurs performances de suivi. De plus, avec cette moyenne, une seule comparaison est réalisée contrairement au premier cas de comparaison élémentaire qui nécessite autant de comparaisons que de poses existant dans le vocabulaire. Cela permet de gagner au niveau complexité de l'algorithme. Ainsi, nous avons opté pour un représentant moyen du vocabulaire pour la pondération des particules.

En résumé, l'avantage du modèle de référence que nous proposons est qu'il est construit d'une manière automatique et il est bien adapté à une cible déformable. Ce modèle de référence va servir ensuite à corriger la trajectoire des particules à travers le processus de pondération. Nous détaillons dans la section suivante les descripteurs de caractérisation de la cible nécessaires à la comparaison des particules avec la référence. Nous décrirons également dans cette section notre contribution principale pour l'amélioration du suivi à travers la pénalisation des particules en utilisant l'information du mouvement extraite avec les flots optiques.

La caractérisation de la cible et la pondération des particules constituent le modèle d'observation que nous décrivons dans la section suivante.

3.2 Le modèle d'observation

Le modèle d'observation détermine la méthode de pondération des particules selon une caractérisation de la cible. Nous proposons d'intégrer différentes catégories de descripteurs pour caractériser la cible. Il est difficile de calculer les poids des particules à partir d'un ensemble d'informations hétérogènes provenant de différentes catégories de descripteurs. Il est nécessaire ainsi de déterminer une méthode de fusion de ces informations. Nous présentons la méthode de fusion des caractéristiques de la cible dans la sous-section 3.2.3.

Dans cette section nous commençons par détailler le modèle géométrique représentant la cible. Ensuite, nous détaillons les descripteurs adoptés pour caractériser la cible. Puis, nous présentons la méthode de fusion des descripteurs pour la pondération des particules. Enfin, afin de contrôler l'orientation des particules et améliorer le suivi avec les filtres particulaires, nous proposons une méthode de pénalisation des particules basée sur une méthodes de caractérisation du mouvement.

3.2.1 Représentation géométrique de la cible

La représentation géométrique de la cible fournit un modèle structurel permettant de cerner la cible dans une image. Il représente la fenêtre dans laquelle les caractéristiques de la cible vont être extraites.

Dans notre cas, le suivi de la main a pour but d'extraire sa trajectoire afin de caractériser le geste. Pour extraire cette trajectoire, une position et une taille de la fenêtre cernant la cible suffisent. Les détails de la forme de la main ne sont pas nécessaires à ce niveau. Ainsi, nous avons adopté une fenêtre de forme rectangulaire pour cerner la cible. Cette fenêtre représente la forme des éléments du vocabulaire de la référence et la forme des particules. En effet, dans notre cas, une particule est un contour rectangulaire noté par le vecteur $Y^i = (x^i, y^i, \omega^i, h^i)$ où $p^i = (x^i, y^i)$ est le coin haut gauche représentant la position de la particule et $z^i = (\omega^i, h^i)$ est sa taille. La figure 3.3 représente un exemple d'estimation de la position de la main dominante avec un filtre particulaire en utilisant une représentation rectangulaire.

3.2.2 Caractérisation de la cible

Pour corriger le trajet des particules, les filtres particulaires se réfèrent à une observation réelle construite par des caractéristiques de l'objet suivi. Étant donné que notre cible, la main dominante, est de type déformable, ses caractéristiques

FIGURE 3.3 – Exemple d'estimation de la position de la main dominante avec un filtre particulaire en utilisant une représentation rectangulaire

doivent être constituées de descripteurs invariants aux déformations. Ces descripteurs vont être exploités pour caractériser la référence et le contenu de chaque particule afin d'évaluer la similarité entre les particules et la cible et les pondérer en fonction de cette similarité.

Pour avoir une caractérisation complète de la cible, il est nécessaire de combiner des descripteurs de types complémentaires et discriminants. Il est nécessaire également que ces descripteurs soient capable de caractériser les spécificités de la cible tel que le caractère déformable. Dans notre cas, la cible est déformable avec un changement continu de forme, mais en même temps ne nécessitant pas une description fine de ses contours. Ainsi, nous avons choisi des descripteurs issus de la couleurs qui sont par nature appropriés aux objets déformables : des histogrammes de couleur, des moyennes de couleurs et une proportion de couleur de la peau. Nous avons également choisi des descripteurs issus de la forme fournissant une description globale de la forme et invariants aux transformations géométriques : les moments de Zernike et les moments de Hu. Comme nous l'avons mentionné dans la sous-section 2.5.3 du chapitre précédent, nous exploitons les descripteurs issus du mouvement pour contrôler l'orientation des particule et non pour caractériser la cible. Nous détaillerons cette composante de contrôle des particules dans la sous-section 3.2.4.

Concernant la représentation formelle de ces descripteurs, nous notons le vecteur d'observation par $X = (d_1, ..., d_K)$ où K est le nombre de familles de descripteurs. Le contenu de chaque particule représente une imagette. De cette imagette, l'instance des K familles de descripteurs est extraite. Ainsi, à chaque particule Y^i, un vecteur de caractéristiques $X^i = (d_1^i, ..., d_K^i)$ est associé. D'autre part, le vocabulaire de la référence est constitué d'un ensemble d'imagettes. De même, de chaque imagette de ce vocabulaire, un vecteur de caractéristiques $X^R = (d_1^R, ..., d_K^R)$ est extrait. Comme indiqué dans la section 3.1, le vecteur de caractéristiques $\bar{X}^R$ représentant de ce vocabulaire est le vecteur moyenne des

vecteurs caractéristiques X^R de toutes les imagettes du vocabulaire. Nous détaillerons dans le sous-section 3.2.3 la méthodes de comparaison des vecteurs X^i et $\bar{X}^R$.

Dans notre cas, le vecteur d'observation X^i a une taille de 66 caractéristiques représentant la couleur, la taille et les contours. Dans la suite, nous détaillons les 66 caractéristiques que nous avons adoptées pour caractériser la cible, la main dominante.

3.2.2.1 Descripteurs issus de la couleur

Nous présentons dans cette sous-section trois descripteurs de couleur : les histogrammes de couleur, un indicateur de couleur globale calculé avec des moyennes et des variances et une proportion de couleur dans la fenêtre d'observation.

La couleur décrite avec ces descripteurs est la couleur de la peau de la main. Pour mieux représenter cette couleur, nous avons choisi l'espace de représentation de couleur HSV. Comme nous l'avons mentionné dans la sous-section 2.5.1 du chapitre précédent, il a été montré que dans l'espace de couleur `HSV` l'information de couleur est mieux reproduite [157]. Il a été également montré que cet espace est discriminant entre la couleur de la peau et le fond et comprend deux composantes (hue et saturation) robustes aux changements de luminosité [125].

L'espace de couleur `HSV` (Hue Saturation Value) est un espace à coordonnées cylindriques comme illustré dans la figure 3.4 [1]. La composante hue définit la teinte de la couleur de l'objet, codée suivant l'angle qui lui correspond sur le cercle des couleurs. Ses valeurs varient entre 0° et 360°. Les composantes value et saturation représentent respectivement un pourcentage d'*intensité* de couleur et de *brillance* de couleur.

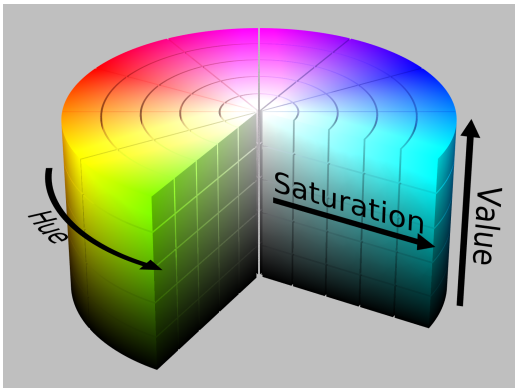


FIGURE 3.4 – Représentation de l'espace HSV

Pour générer les descripteurs de la couleur de la peau, il est nécessaire de définir un modèle de la couleur de la peau. Il est possible d'appliquer un proces-

1. http ://en.wikipedia.org/wiki/HSL_and_HSV

sus d'apprentissage pour générer ce modèle. Cependant, pour éviter le problème d'étiquetage d'une base d'apprentissage et pour généraliser la méthode de détermination de la couleur de la peau, nous avons conçu un processus automatique de détection de cette couleur dans la vidéo traitée. Cet algorithme permet d'obtenir un modèle de couleur de la peau du signeur dans la vidéo sous forme de trois histogrammes de couleur selon l'espace HSV.

L'idée principale de cet algorithme est la détection du visage de l'acteur. En effet, nous appliquons en premier lieu la méthode de Viola et Jones [139] pour détecter le visage du signeur. Cette méthode est connue pour son efficacité pour la détection des visages. Nous extrayons ensuite, de l'imagette du visage détecté, les trois histogrammes correspondant aux trois dimensions *hue*, *saturation* et *value* de l'espace HSV. Puis, nous calculons la moyenne pondérée (équation 3.1) et la variance correspondante pour chaque histogramme. m est la moyenne des valeurs quantifiées de l'histogramme ($\left[(b+\frac{1}{2})\times\frac{H_{max}}{Nbins}\right]$) pondérées par leurs fréquences ($H(b)$). Cette moyenne et la variance associée permettent de cerner plus précisément l'intervalle de la couleur de la peau parmi les valeurs de la composante de l'espace de couleurs. Cette opération de filtrage permet de construire un masque de couleur de peau facilitant son extraction sur une imagette quelconque et assurant la robustesse de la méthode à la variabilité des signeurs. Pour des raisons de précision, nous gardons la totalité de cet histogramme masque dans le vecteur de caractéristiques au lieu de la moyenne et la variance. Ainsi, nos deux premiers descripteurs sont des histogrammes de couleur de peau selon les deux composantes *hue*, *value*. Ils forment un vecteur de taille $48 = 16(hue)+32(value)$. L'histogramme de la composante *value* n'est pas pris en compte car cette composante dépend fortement de la variation de la luminosité dans l'environnement, or notre but est de construire un vecteur de caractéristiques invariant aux conditions d'enregistrement de la vidéo.

$$m = \frac{1}{\sum_{b=1}^{Nbins} H(b)} \sum_{b=1}^{Nbins} H(b) \times \left[(b+\frac{1}{2})\times\frac{H_{max}}{Nbins}\right] \tag{3.1}$$

En ce qui concerne le vecteur des caractéristiques du représentant du vocabulaire de la référence présenté dans le paragraphe 3.1, les descripteurs des histogrammes de couleur sont décrits par les histogrammes moyens calculés sur l'ensemble des éléments du vocabulaire.

Nous ajoutons quatre caractéristiques génériques de couleur donnant une indication sur la couleur globale dans l'imagette. Ils sont sous forme d'une moyenne normalisée de la teinte (*hue*) et la variance correspondante et une moyenne normalisée de la saturation et la variance correspondante sur tous les pixels de l'imagette. La moyenne et la variance de la composante *value* ne sont pas prises en compte puisque nous voulons toujours vérifier la robustesse à la variabilité de la luminosité comme expliqué auparavant.

Notre troisième type de descripteur de couleur est une caractéristique de taille relative. Cette caractéristique représente la proportion de la zone de peau dans l'imagette de la particule. Avec la méthode `Camshift` expliquée dans la sous-section A.2 de l'annexe A, nous calculons la taille de la zone de la peau dans l'imagette en prenant comme sélection de départ l'imagette de la particule en totalité. D'une manière indirecte, ce descripteur permet de contrôler la taille de la particule en s'assurant de conserver une certaine proportion de peau dans les particules. Il permettra de favoriser les particules qui délimitent l'objet en couleur de peau et ne contenant pas une grande partie du fond ou d'autres objets dont la couleur est différente de la couleur de la peau. La situation où la particule contient deux ou plusieurs objets différents de la cible mais ayant une couleur de peau est censée être contrôlée par les autres caractéristiques complémentaires et la méthode de pénalisation.

3.2.2.2 Descripteurs issus de la forme

Afin de décrire la forme de la main tout en restant générique, nous ajoutons des descripteurs de contours connus pour leur invariance à la rotation, à la translation, au changement d'échelle et à la symétrie axiale comme indiqué dans la sous-section 2.5.2. Ces descripteurs sont les moments de `Hu` , notés par $h_i, i = 1..7$, et les moments de `Zernike`. Les magnitudes des moments de zernike sont notées par $Z_{p,q}$ et sont définies par un ordre p et une répétition q sur l'ensemble $\{(p,q) \in \mathbb{N} \times \mathbb{Z}, |q| \leq p, p - |q| = 2k, k \in \mathbb{N}\}$. Une étude établie dans [80] montre que les 21 premiers moments de zernike peuvent être classés selon des relations de corrélation en 6 classes représentées par $Z_{4,0}$, $Z_{5,1}$, $Z_{4,2}$, $Z_{5,3}$, $Z_{4,4}$, $Z_{5,5}$.

Ainsi, notre vecteur de descripteurs de contours comprend 13 valeurs composées de 7 moments de `Hu` et de 6 moments de `zernike` choisis selon les travaux de Lin [80].

Le tableau 3.1 résume les sous-vecteurs de descripteurs adoptés pour la caractérisation de la cible, leurs tailles, leurs domaines de variation où $taille_f$ est la taille de la fenêtre rectangulaire représentant la cible, et les notations des distances de comparaison avec la référence qui seront présentées dans la sous-section 3.2.3.1. Ce tableau montre que ces descripteurs sont de divers natures et de divers ordres de valeurs nécessitant ainsi une méthode de fusion des informations. Dans la sous-section suivante, nous allons montrer comment nous avons fusionné ces informations et comment nous les avons intégrées dans une probabilité d'observation permettant de pondérer les particules.

3.2.3 Estimation de la probabilité d'observation

Dans ce paragraphe nous allons montrer comment l'information apportée par les descripteurs, décrits précédemment, est introduite dans le mécanisme de suivi

	descripteurs issus de la couleur			**descripteurs issus de la forme**	
	histogrammes de couleur	moyennes et variances de couleur	proportion de couleur de la peau	moments de Hu	moments de Zernike
taille du vecteur	$48 = 16(hue) + 32(value)$	$4 = 2moyennes + 2variances$	1	7	6
domaine de variation	$[0, taille_f]$	$[0, \infty[$	$[0, 1]$	$[0, 1]$ $\propto 10^{-n}, n \in \mathbb{N}$	$[-1, 1]$
Distance de comparaison	distance de Bhattacharayya : D_1	distance euclidienne : D_5	distance euclidienne : D_4	distance spécifique au moments de Hu : D_2	distance euclidienne : D_3

TABLE 3.1 – Bilan sur les descripteurs adoptés pour la caractérisation de la cible

du filtre particulaire.

La mise à jour des poids est une forme de correction de la prédiction du filtre particulaire. La correction pondère les particules selon leur proximité de la référence en calculant une distance entre une caractérisation de l'observation associée X^i dans la particule Y^i et une caractérisation $\bar{X}^R$ de la référence R. La caractérisation de la particule et la caractérisation de la référence sont réalisées à l'aide des descripteurs définis dans le paragraphe 3.2.2. Puisque ces descripteurs présentent différentes modalités et différents domaines de variation, il est difficile de les fusionner au préalable et de leur appliquer une seule distance. Il s'avère nécessaire ainsi de spécialiser les distances selon les catégories de descripteurs. Dans ce cas, la fusion est appliquée sur les distances en les intégrant dans la probabilité d'observation $P(X|Y)$. Dans le cas où les descripteurs ne sont pas fusionnés au préalable, les chercheurs choisissent en général la forme gaussienne pour cette probabilité [71]. Dans notre cas, à travers nos expérimentations, nous avons constaté que la forme 3.2 donne de meilleurs résultats de suivi.

$$w^i = P(X^i|Y^i) = \prod_{l=1}^{K} \left(\frac{1}{1 + D_l(X^i, \bar{X}^R)} \right)^{c_l} \tag{3.2}$$

Pour chaque famille de descripteurs l, une distance D_l est définie. Ainsi, pour chaque famille de descripteurs l, une probabilité élémentaire d'observation est calculée sous la forme $\left(\frac{1}{1+D_l(X^i,\bar{X}^R)}\right)^{c_l}$ où c_l est un coefficient puissance de pondération de la famille de descripteurs. Ce coefficient permet de mettre en valeur les familles de descripteurs les plus pertinentes. Enfin, la probabilité d'observation globale est le produit des K probabilités élémentaires associées aux familles de descripteurs comme le montre l'équation 3.2.

Nous détaillons dans le paragraphe suivant les distances D_l qui varient selon la catégorie de descripteurs associée.

3.2.3.1 Choix des distances appliquées aux descripteurs

La distance D_1 est associée aux histogrammes de couleur (équation 3.3). Plusieurs fonctions de comparaison des histogrammes sont proposées dans la littérature : la distance de corrélation, la distance d'intersection, la distance Chi-deux, la distance de Bhattacharayya et la distance EMD (« *Earth Mover's Distance* ») [19, 16, 118]. Dans notre cas, nous comparons deux histogrammes de deux imagettes de taille différente. Ainsi, pour avoir une distance indépendante de la taille des imagettes, il est nécessaire de normaliser les histogrammes. La méthode de comparaison de Bhattacharayya [16] ajoute une étape de normalisation des histogrammes. L'équation 3.4 présente la distance de Bhattacharayya. Nous appliquons alors cette distance pour comparer les histogrammes de couleurs des particules au modèle de la référence. Les valeurs de cette distance sont comprises entre 0 et 1. Plus les histogrammes sont similaires, plus la distance de Bhattacharayya tend vers 0.

$$D_1(X^1, X^2) = \frac{1}{2} D_{Bhat}(H_1^{hue}, H_2^{hue}) + \frac{1}{2} D_{Bhat}(H_1^{saturation}, H_2^{saturation}) \tag{3.3}$$

$$D_{Bhat}(H_1, H_2) = \sqrt{1 - \sum_b \frac{\sqrt{H_1(b).H_2(b)}}{\sqrt{\sum_b H_1(b). \sum_b H_1(b)}}} \tag{3.4}$$

De même, il existe différentes distances adaptées aux moments de `Hu` présentées dans [19]. D'après nos expérimentations, nous avons constaté que la formule de comparaison présentée par l'équation 3.5 est la moins sensible aux différences qui peuvent exister entre deux images représentant le même objet.

$$D_2(X^1, X^2) = D_{Hu}(\mathtt{i}_1, \mathtt{i}_2) = \sum_{h=1}^{7} |\frac{1}{m_{\mathtt{i}_1}(h)} - \frac{1}{m_{\mathtt{i}_2}(h)}| \tag{3.5}$$

où $m_{\mathtt{i}_j}(h) = sign(h).log(h)$ et $\mathtt{i}_j$ est l'imagette contenue dans la particule Y^j.

La distance D_3 correspond aux moments de Zernike. Dans le cas des données 2D, la distance classique appliquée pour comparer les moments de Zernike est en général la distance euclidienne [114]. Cette distance permet de conserver l'invariance à la rotation et l'invariance à la symétrie axiale [114]. Ainsi, dans notre cas, la distance D_3 correspond à la distance euclidienne.

Les descripteurs de proportion, des moyennes et des variances de la couleur de peau sont des caractéristiques simples de type scalaire. Nous avons testé la distance de Manhattan et la distance euclidienne pour comparer ces caractéristiques et nous avons constaté que la distance euclidienne donne de meilleurs résultats

de suivi. Ainsi, les distances D_4 et D_5 représentent la distance euclidienne. La distance D_4 correspond au descripteur de proportion de couleur de peau et la distance D_5 correspond au descripteur des moyennes et des variances de la couleur de peau. la distance D_5 compare deux vecteurs de scalaires.

Le modèle d'observation que nous venons de proposer est construit avec des caractéristiques statiques qui décrivent la cible. Puisque le but est de suivre une cible mobile, il est intéressant également d'exploiter les caractéristiques de son mouvement afin d'améliorer son suivi. Néanmoins, le mouvement n'est pas une caractéristique absolue de la cible mais dépend de son état. Par exemple, il n'est pas possible de décrire la référence des filtres particulaires avec une caractéristique dynamique comme le mouvement. La description de la référence doit être générique, absolue et indépendante de la situation analysée. En d'autres termes, la caractérisation de la référence ne doit pas dépendre de l'état statique ou mobile de la cible. Ainsi, nous avons introduit l'information de mouvement différemment dans le processus de suivi. Nous l'avons introduite sous forme d'une pénalisation dans l'étape de mise à jour des poids des particules et dans le modèle de leur mouvement.

3.2.4 Pénalisation des particules avec les flots optiques

Dans le cas où la cible a un mouvement rapide, le risque de divergence des filtres particulaires classiques augmente. Pour diminuer ce risque et orienter les particules vers les zones les plus probables, nous proposons de pénaliser les particules qui se déplacent contre le flot observé. Nous réalisons cette pénalisation à l'aide des informations extraites avec les flots optiques présentés dans la sous-section 2.5.3. Nous rappelons que les flots optiques sont une méthode de calcul des vitesses des pixels, connus pour leur robustesse aux variations de luminosité et aux déformations de l'objet suivi [15]. L'observation du flot s'effectue localement à l'intérieur de chaque particule. La vitesse de la particule et la vitesse de son flot interne sont considérées indépendantes et la pénalisation se base sur la comparaison de ces deux vitesses. Plus l'écart entre ces deux vitesses est grand, plus le poids est affaibli.

En effet, nous calculons d'abord une carte de vitesses Ψ_t à partir des deux images successives $\mathtt{i}_t$ et $\mathtt{i}_{t-1}$ en appliquant la méthode d'extraction des flots optiques de Lucas-Kanade [85]. Ensuite, dans chaque particule Y_t^i, nous calculons la médiane $\vec{\nu}_t^i$ des vitesses des pixels inscrits dans la particule. D'autre part, chaque particule a un vecteur de déplacement que nous notons $\vec{p}_t^i$. Nous évaluons ensuite la distance entre ces deux vecteurs, $\vec{\nu}_t^i$ et $\vec{p}_t^i$, que nous notons ξ_t^i (équation 3.6), et nous introduisons cette distance dans le calcul du poids de la particule. L'introduction de cette pénalisation se fait en multipliant le poids w_t^i par le terme ξ_t^i et nous notons le nouveau poids par $w_t'^i = w_t^i \xi_t^i$. La pénalisation avec le terme ξ_t^i représente une deuxième étape de mis à jour des poids après l'étape classique de mise à jour

des poids définie dans l'algorithme de ConDensAtion (algorithme 5) qui génère le poids w_t^i. Les particules vont être ensuite échantillonnées selon les nouveaux poids $w_t'^i$.

$$\xi_t^i = \frac{1}{1+\lambda_t^i}[cos(\widehat{\vec{\nu}_t^i \delta t, \vec{p}_t^i})]^{\tau_t^i}. \tag{3.6}$$

L'équation 3.6 représente la forme du terme ξ_t^i correspondant à la particule Y_t^i. Les deux termes λ_t^i et τ_t^i sont définis dans le tableau 3.2 selon des conditions sur les deux vecteurs $\vec{\nu}_t^i$ et $\vec{p}_t^i$.

Le tableau 3.2 résume les situations simultanées possibles de l'orientation de la particule et de l'orientation de son contenu. La première situation représente le cas où la particule et son contenu sont stationnaires, les deux entités ont un état similaire et donc la pénalisation doit être nulle et sans effet sur le poids. La deuxième situation représente le cas où la particule ou son contenu est stationnaire tout en excluant la situation 1. Dans ce cas, l'état des deux entités est considéré complètement opposé et donc la pénalisation doit être maximale. La constante $\Lambda \in \mathbb{R}^+$ est une valeur à déterminer empiriquement et doit être suffisamment élevée pour faire tendre ξ_t^i vers 0. Pour la troisième et la quatrième situation, la particule et son contenu ont chacun un mouvement non nul, il reste alors à comparer les orientations. Si les deux entités ont des orientations formant un angle obtus (condition 3), alors le sens de leurs mouvements est considéré suffisamment différent pour les pénaliser fortement. Si les orientations des deux entités forment un angle aigu (condition 4), alors les deux orientations sont considérées similaires et la pénalisation dépend de l'écart entre les deux directions. Le terme de pénalisation ξ_t^i est compris entre 0 et 1, plus l'écart entre les deux directions est important, plus la valeur de ξ_t^i est réduite et plus l'effet de la pénalisation est important. Le terme τ_t^i contrôle la prise en compte du terme $cos(\widehat{\vec{\nu}_t^i \delta t, \vec{p}_t^i})$, il est à 1 dans ce cas. Le terme λ_t^i dépend, dans cette situation, de la différence de rapidité du déplacement des deux entités.

Conditions	condition 1	condition 2	condition 3	condition 4
	$\|\vec{\nu}_t^i \delta t\| = 0$ AND $\|\vec{p}_t^i\| = 0$	$\|\vec{\nu}_t^i \delta t\| = 0$ XOR $\|\vec{p}_t^i\| = 0$	$\|\vec{\nu}_t^i \delta t\| \|\vec{p}_t^i\| \neq 0$	
			$cos(\widehat{\vec{\nu}_t^i \delta t, \vec{p}_t^i}) \leq 0$	$cos(\widehat{\vec{\nu}_t^i \delta t, \vec{p}_t^i}) > 0$
(λ_t^i, τ_t^i)	$(0, 0)$	$(\Lambda, 0)$	$(\Lambda, 0)$	$(\|\|\vec{\nu}_t^i \delta t\| - \|\vec{p}_t^i\|\|, 1)$

TABLE 3.2 – Les valeurs de λ_t^i et de τ_t^i selon les situations simultanées possibles de l'orientation de la particule et de l'orientation de son contenu

Ainsi, nous appliquons un contrôle de l'orientation des particules en exploitant l'information de vitesse extraite avec les flots optiques. Ce contrôle va être explicité davantage sur un exemple pratique avec la figure 3.7 dans le paragraphe 3.5.

Conjointement à ce contrôle local des particules, nous introduisons un contrôle global du déplacement des particules en exploitant la même information extraite avec les flots optiques. Afin de lisser le mouvement des particules, nous ajoutons un terme de vitesse globale de l'estimation du filtre particulaire au modèle du mouvement des particules en le pénalisant avec une vitesse globale du flot dans toute la scène. Nous expliquons davantage cette deuxième contribution dans la section suivante.

3.3 Le modèle de mouvement des particules

Nous avons agi sur le modèle de propagation des particules en introduisant des termes non-linéaires afin de lisser le mouvement des particules et de l'adapter à la rapidité du déplacement de l'objet suivi. Nous avons introduit dans le modèle de mouvement la vitesse et l'accélération du représentant des particules $\hat{Y}$. Pour régulariser ces termes de lissage et garder une liaison avec l'observation réelle, nous avons pondéré ces termes par des coefficients proportionnels à des informations du mouvement global issues des flots optiques. Le mouvement global peut être dans certains cas largement influencé par un mouvement dominant d'un seul objet mobile dans la scène comme dans le cas de la langue des signes (main dominante).

L'équation 3.7 représente le modèle classique de mouvement des particules selon l'algorithme de ConDensAtion.

$$Y_t^i = AY_{t-1}^i + BR_t^i. \tag{3.7}$$

où A est la matrice de transition, R_t^i est un vecteur aléatoire et B est la matrice de la marche aléatoire. Dans notre cas, A et B sont constantes.

Le modèle classique de mouvement disperse les particules aux alentours de leur ancienne position sans orientation bien précise. Cette dispersion aléatoire ralentit le mouvement des particules et rend difficile le suivi des mouvements rapides.

$$Y_t^i = AY_{t-1}^i + BR_t^i + \alpha_t \begin{bmatrix} \dot{\hat{p}}_{t-1} \\ 0 \\ 0 \end{bmatrix} + \beta_t \begin{bmatrix} \ddot{\hat{p}}_{t-1} \\ 0 \\ 0 \end{bmatrix} \tag{3.8}$$

Pour lisser et harmoniser le mouvement des particules, nous proposons d'introduire la vitesse, $\dot{\hat{p}}_{t-1}$, et l'accélération, $\ddot{\hat{p}}_{t-1}$, de l'estimation du filtre dans le modèle de mouvement des particules (équation 3.8). Comme indiqué dans l'algorithme 5 de ConDensAtion, cette estimation a la forme $\hat{Y}_t = \sum_{i=1}^{N} \tilde{w}_t^{(i)} f(Y_t^{(i)})$, où $\tilde{w}_t^{(i)}$ sont les poids normalisés. En affectant l'identité à la fonction f, l'estimation devient la moyenne pondérée des particules où la pondération est réalisée avec les poids des particules. Par conséquent, plus les poids sont distincts et représentatifs de la proximité à la cible, plus l'estimation est considérée correcte et plus le degré de prise en compte des informations de mouvement de cette estimation peut être

important. Nous avons traduit ce principe par l'introduction de deux coefficients α_t et β_t qui régularisent l'effet de la vitesse et de l'accélération de l'estimation dans le modèle de mouvement des particules (équation 3.8). Ces coefficients peuvent être des constantes déterminées expérimentalement comme pour le cas du système `2VPF` qui sera expliqué dans le paragraphe 3.4.3.

Pour réduire le risque d'introduction de l'erreur récursive de l'estimation dans le modèle de mouvement, nous avons conçu également une forme variable de ces coefficients. En effet, ces deux coefficients peuvent dépendre d'une vitesse globale qui donne une indication sur l'intensité du mouvement dominant. S'il existe un mouvement important entre deux images, ces coefficients vont avoir une valeur élevée et par la suite le déplacement des particules va être élevé et peut suivre le flot. Si le mouvement global est faible, ces coefficients vont avoir une valeur réduite et par la suite le mouvement des particules va être freiné, cette action évitera que le filtre diverge.

Dans le cas où ces coefficients α_t et β_t sont variables, ils dépendent d'une vitesse moyenne normalisée comme suit :

$$\alpha_t = \begin{pmatrix} \frac{\bar{\vartheta}(\Psi_t^x)}{\max_j \vartheta^j(\mathtt{s}^x)} & 0 & 0 & 0 \\ 0 & \frac{\bar{\vartheta}(\Psi_t^y)}{\max_j \vartheta^j(\mathtt{s}^y)} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \beta_t = \begin{pmatrix} \frac{\bar{\gamma}(\Psi_t^x)}{\max_j \gamma^j(\mathtt{s}^x)} & 0 & 0 & 0 \\ 0 & \frac{\bar{\gamma}(\Psi_t^y)}{\max_j \gamma^j(\mathtt{s}^y)} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Ψ_t est la carte des vitesses calculées avec les flots optiques qui sont calculés à partir des deux images $\mathtt{i}_t$ et $\mathtt{i}_{t-1}$. $\vartheta^j(\Psi_t^x)$ (resp. $\vartheta^j(\Psi_t^y)$) est la valeur absolue de la composante horizontale (resp. verticale) de la vitesse du pixel j. $\bar{\vartheta}(\Psi_t^x)$ (resp. $\bar{\vartheta}(\Psi_t^y)$) est la moyenne des valeurs absolues des vitesses horizontales (resp. verticales) sur toute la carte Ψ_t^x (resp. Ψ_t^y). Cette vitesse moyenne horizontale (resp. verticale) est normalisée par une vitesse maximale horizontale (resp. verticale) sur toute la séquence.

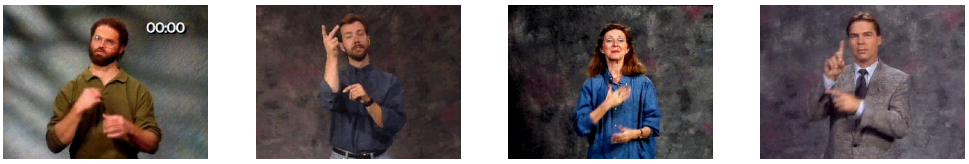
Nous montrons dans le prochain paragraphe l'intérêt de nos contributions à travers notre expérimentation réalisée sur des vidéos de discours de la langue des signes.

3.4 Le protocole expérimental

3.4.1 Les données expérimentales

Nous avons élaboré l'expérimentation du suivi de la main dominante sur une base de données de langue des signes américaine nommée SignStream-ASLLRP [97]. Cette base consiste en quatre séquences $\mathtt{s}_1$, $\mathtt{s}_2$, $\mathtt{s}_3$ et $\mathtt{s}_4$ de discours en langue des signes de taille importante. Chaque séquence de vidéo contient entre 1310 et 5046 trames, acquises dans un studio d'enregistrement. La fréquence des

trames est entre 30 et 32 trames par seconde. La taille d'une trame est entre 288×216 et 320×240 pixels. Chaque vidéo contient un seul signeur, qui signe un discours de la langue des signes face à une caméra fixe sur un font presque uni. Chaque vidéo de la base SignStream contient un signeur différent qui joue un discours différent. Comme le montre la figure 3.5, il n'y a aucune contrainte sur le style ou la couleur des vêtements des quatre signeurs.



(a) trame 34 de la séquence $\mathtt{s}_1$ (b) trame 333 de la séquence $\mathtt{s}_2$ (c) trame 66 de la séquence $\mathtt{s}_3$ (d) trame 51 de la séquence $\mathtt{s}_4$

FIGURE 3.5 – Exemple de trames de chaque séquence de la base SignStream-ASLLRP

À cause du problème de l'annotation, nous avons effectué le réglage des paramètres de notre système en utilisant $\mathtt{s}_1$, $\mathtt{s}_2$, et $\mathtt{s}_3$ et nous avons exploité uniquement la séquence $\mathtt{s}_4$, qui compte 1310 trames, pour l'évaluation du système. En effet, nous avons constitué une vérité terrain sur la vidéo $\mathtt{s}_4$ en encadrant manuellement dans chaque trame la cible, qui est la main dominante. L'évaluation du suivi pour le réglage des paramètres a été réalisée visuellement. Si nous avions pu disposer d'assez de données annotées, nous aurions pu évaluer notre système sur les quatre vidéos en utilisant la méthode de validation croisée. Nous expliquons dans le paragraphe suivant, comment nous comparons l'estimation du filtre particulaire $\hat{Y}$ avec la position réelle de la cible $\mathtt{G}$.

3.4.2 Les métriques d'évaluation

Comme expliqué dans la section 2.6 du chapitre précédent, les principales méthodes utilisées pour l'évaluation du suivi des objets dans des vidéos dans la littérature sont :

- l'évaluation visuelle à travers la représentation de l'estimation sur les trames,
- le calcul d'une erreur de suivi sous forme d'une distance en pixels entre la position réelle de la cible et la position estimée,
- le calcul du taux de suivi issu de l'indice de Jaccard.

Afin d'évaluer précisément la qualité du suivi réalisé, nous avons adopté ces trois méthodes d'évaluation.

Nous calculons l'erreur totale de suivi $\bar{\epsilon}$ dans la vidéo en utilisant la moyenne des erreurs élémentaires de position (équation 3.9). Dans [11], cette erreur n'est calculée que lorsqu'il y a intersection entre la région estimée de l'objet et la région

réelle de l'objet. Dans notre cas, nous calculons cette erreur sur toutes les trames de la vidéo quelle que soit la position des deux entités comparées. Nous estimons que cette forme d'erreur donne une information plus précise sur le suivi.

$$\bar{\epsilon} = \frac{1}{|\mathtt{s}|} \sum_{t=1}^{|\mathtt{s}|} ||\hat{p}_t - \mathtt{G}_{p,t}|| \tag{3.9}$$

où $|\mathtt{s}|$ est le nombre de trames d'une séquence $\mathtt{s}$, $\hat{p}_t$ est la position estimée de la cible $\mathtt{G}$ à l'instant t et $\mathtt{G}_{p,t}$ est sa position réelle à l'instant t. $\mathtt{G}_{p,t}$ correspond au coin haut gauche du cadre délimitant la cible.

Le taux total de suivi $\bar{r}$ est représenté par une moyenne des taux élémentaires sur toutes les trames de la vidéo (équation 3.10). Nous rappelons que ce taux de suivi moyen permet de classer la qualité de suivi selon quatre catégories : « suivi médiocre », « suivi partiel », « suivi globalement satisfaisant » et « suivi excellent » (tableau 2.1). Dans notre cas, nous avons utilisé ϱ_t (équation 3.12) qui est une métrique issue de l'application directe de l'indice de Jaccard (équation 3.11) et que nous appelons l'indice de superposition. Pour pourvoir classer la qualité de suivi de nos systèmes, nous avons écrit le taux de suivi r_t en fonction de l'indice de superposition ϱ_t selon l'équation 3.13. Les résultats présentés dans la section 3.5 correspondent aux moyennes de ces métriques sur toute la séquence de test.

$$r_t = \frac{2 \times S(\hat{Y}_t \cap \mathtt{G}_t)}{S(\hat{Y}_t) + S(\mathtt{G}_t)} \tag{3.10}$$

$$\varrho_t = J(\hat{Y}_t, \mathtt{G}_t) = \frac{|\hat{Y}_t \cap \mathtt{G}_t|}{|\hat{Y}_t \cup \mathtt{G}_t|} \tag{3.11}$$

$$\varrho_t = \frac{S(\hat{Y}_t \cap \mathtt{G}_t)}{S(\hat{Y}_t) + S(\mathtt{G}_t) - S(\hat{Y}_t \cap \mathtt{G}_t)} \tag{3.12}$$

$$r_t = \frac{2\varrho_t}{1 + \varrho_t} \tag{3.13}$$

3.4.3 Les systèmes évalués

Dans le but d'évaluer la robustesse de notre système et montrer la contribution de chacune de ses composantes, nous comparons quatre configurations du filtre particulaire : le filtre `PF`, le filtre `VPF`, le filtre `2VPF`, et le filtre `3VPF`. Chaque système est une version augmentée du précédent.

Le filtre `PF` est la version classique du filtre particulaire. Le modèle d'observation que nous avons utilisé est celui présenté dans le paragraphe 3.2.

Le filtre `VPF` est un filtre `PF` dont le modèle de référence est construit avec notre proposition de vocabulaire de référence.

Le filtre 2VPF intègre la vitesse et l'accélération de l'estimation $\hat{Y}$ dans le modèle de mouvement des particules du filtre VPF comme nous l'avons proposé dans le paragraphe 3.3. Pour ce système 2VPF, les coefficients α_t et β_t sont des constantes déterminées expérimentalement.

Enfin, le filtre 3VPF présente le système complet avec toutes nos contributions. En effet, le filtre 3VPF est le filtre 2VPF avec une intégration de la pénalisation locale, au niveau des coefficients α_t et β_t (paragraphe 3.3), et globale, au niveau de la mise à jour des poids (paragraphe 3.2.4), conçue avec les flots optiques.

Les paramètres communs entre les quatre systèmes sont : le nombre de particules $\mathcal{N} = 100$, la matrice A est la matrice identité, et la matrice B et le coefficient c_l sont déterminés expérimentalement.

3.5 Résultats de suivi

système	PF	VPF	2VPF	3VPF
$0<\bar{\epsilon}<400$	54.28	31.82	29.23	21.22
$0<\frac{\bar{\epsilon}}{400}<1$	0.136	0,079	0,073	0,053
$0<\bar{\varrho}<1$	0.004	0.279	0.315	0.369
$0<\bar{r}<1$	0.007	0.386	0.435	0.505

TABLE 3.3 – L'évolution de l'erreur moyenne du suivi ($\bar{\epsilon}$), de l'indice moyen de superposition ($\bar{\varrho}$) et du taux moyen du suivi ($\bar{r}$) pour les quatre systèmes PF, VPF, 2VPF et 3VPF sur la séquence $\mathtt{s}_4$ composée de 1310 trames

Le tableau 3.3 montre l'évolution de l'erreur moyenne du suivi ($\bar{\epsilon}$) et l'évolution de l'indice moyen de superposition ($\bar{\varrho}$) avec chaque contribution. Cette évaluation est effectuée sur la séquence $\mathtt{s}_4$ composée de 1310 trames. Les performances du système du suivi augmentent avec la diminution de l'erreur moyenne du suivi $\bar{\epsilon}$ et l'augmentation de l'indice moyen de superposition $\bar{\varrho}$. Dans le cas où les trames de la vidéo de test sont de taille 320×240, l'erreur moyenne $\bar{\epsilon}$, calculée en nombre de pixels, est bornée par la valeur 400 qui correspond à la distance la plus grande entre deux points de l'image (diagonale de l'image). Cette valeur maximale permet de relativiser les valeurs de cette métrique $\bar{\epsilon}$. Selon l'équation 3.12, la métrique $\bar{\varrho}$ est un pourcentage borné par 1.

La valeur de $\bar{\varrho}$ est considérablement faible pour le système PF car le filtre est totalement attiré par le visage et parfois attiré par la main dominée lors de son passage à proximité de la main dominante qui est la vraie cible. Cette distraction est dû principalement à la faiblesse du modèle de référence classique qui est exprimé sous la forme de la pose initiale de la cible. Le saut de performance entre le système PF et le système VPF prouve l'apport du vocabulaire de référence.

L'évolution des valeurs des deux métriques $\bar{\epsilon}$ et $\bar{\varrho}$ exposées dans le tableau 3.3 montre l'apport de chacune de nos contributions au niveau de l'exactitude

du suivi. Le système VPF renforce l'attache aux données à travers le modèle de référence générique. Le système 2VPF a l'avantage de faciliter le suivi des mouvements rapides à travers le lissage du mouvement des particules. Enfin, le système 3VPF permet de limiter l'espace d'exploration des particules aux zones pouvant le plus probablement contenir l'objet en se basant sur des informations de mouvement.

Comme le montre le tableau tableau 3.3, la valeur du taux moyen de suivi $\bar{r}$ correspondant au système complet 3VPF est entre à 0.4 et 0.7. Ainsi, en se référant au tableau 2.1 du classement de la qualité du suivi des systèmes de suivi, nous déduisons que le suivi de notre système complet 3VPF est un « `suivi partiel` ».

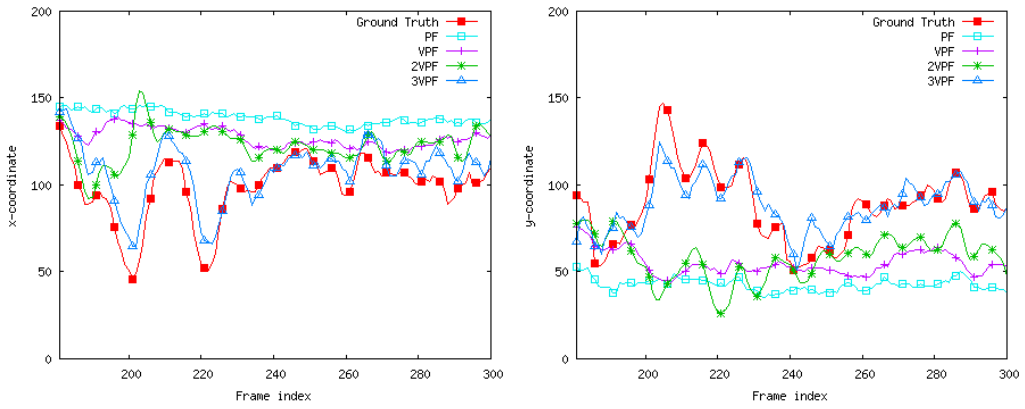


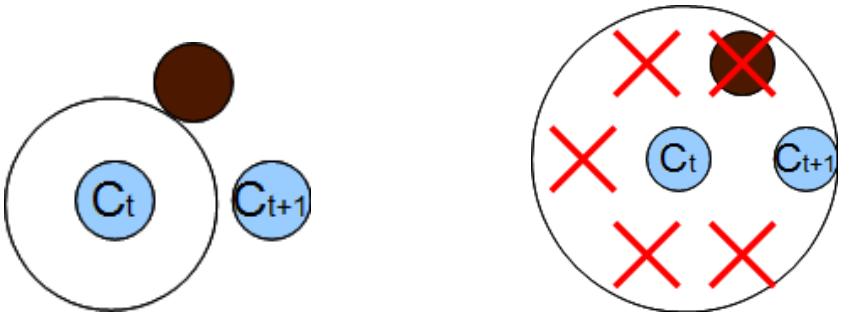
FIGURE 3.6 – La variation des coordonnées x et y de l'estimation du filtre particulaire pour les quatres systèmes PF, VPF, 2VPF et 3VPF sur un extrait de la vidéo s_4 contenant uniquement 120 trames pour des raisons de clarté

La figure 3.6 donne la variation des coordonnées x et y de l'estimation du filtre particulaire pour les quatres systèmes PF, VPF, 2VPF et 3VPF sur un extrait de la vidéo s_4. Pour des raisons de clarté, cet extrait contient uniquement 120 trames entre les trames 180 et 300. Cette figure permet d'expliciter les résultats de l'erreur moyenne de suivi $\bar{\epsilon}$ pour les quatre systèmes dans le cas de mouvements rapides et irréguliers. Nous pouvons voir sur cette double figure le déplacement horizontal et vertical des quatre systèmes par rapport au déplacement réel de la cible. Il est clair que le système 3VPF donne l'estimation la plus proche de la trajectoire réelle. De plus, ce système est capable de suivre les mouvements rapides et irréguliers de la cible et les changements brusques de sa trajectoire comme le montre la figure 3.6 au niveau des trames 200 et 220. Le système 2VPF est capable de suivre ces changements brusques grâce au modèle de mouvement raffiné mais peut s'égarer facilement de la cible. Cependant, le système 3VPF limite la zone de déplacement des particules ce qui réduit considérablement ce risque. Par rapport au système classique PF, le système VPF génère une trajectoire plus proche à la trajectoire réelle mais les trajectoires des estimations des deux systèmes PF et VPF restent

monotones et relativement loin de cette trajectoire réelle.

Le tableau 3.4 illustre concrètement le comportement de suivi de chaque système sur les trames 198, 206, 214, 222 et 230 correspondant aux points de variation aigüe de la trajectoire réelle de la cible présentée dans la figure 3.6. Le cadre magenta correspond à l'estimation $\hat{Y}$ représentant l'ensemble des particules générées par le système particulaire. La cible est la main droite du signeur, elle correspond à la main dominante. Il est clair que l'estimation du système `PF` est totalement attirée par le visage et l'estimation du système `VPF` reste stable et n'arrive pas à suivre le mouvement rapide de la main. L'estimation du système `2VPF` saute entre la main et le visage mais ne reste pas "accrochée" à la main. Nous pouvons expliquer ce phénomène par la dispersion des particules entre la main et le visage. Les particules ont une liberté d'exploration large qui peut avoir un effet négatif sur l'exactitude du suivi et peut causer l'éloignement du filtre de sa cible. Le mécanisme de pénalisation du système `3VPF` vient corriger ce défaut en cernant la zone de déplacement des particules selon des conditions de mouvement bien précises (tableau 3.2). Dans cette situation, on peut se demander pourquoi ne pas réduire le pas de la marche aléatoire représentée par la matrice B dans le modèle de mouvement des particules. En réalité, si le pas de la marche aléatoire est réduit, les particules risquent de ne pas retrouver la cible dans l'étape suivante. La figure 3.7 illustre ce phénomène de taille de la marche aléatoire d'une manière simplifiée. Le disque C_t représente la position de la cible à l'instant t, le disque C_{t+1} représente la position de la cible à l'instant $t+1$. Le disque foncé représente un objet qui a des caractéristiques similaires aux caractéristiques de la cible et a un mouvement négligeable. Le grand cercle représente la marche aléatoire. Pour éviter l'attraction du filtre par l'objet similaire à la cible, deux solutions se présentent. Une première solution est de réduire la taille de la marche aléatoire comme dans la figure 3.7a de telle sorte que l'objet causant la confusion quitte l'espace d'exploration des particules. Cependant, avec un déplacement important de la cible à l'instant $t+1$, cette dernière sort également de l'espace d'exploration des particules. Ainsi, le filtre peut facilement diverger. Une deuxième solution est de garder un grand pas de la marche aléatoire et d'appliquer en même temps le mécanisme de pénalisation avec les flots optiques comme expliqué dans la sous-section 3.2.4. Avec le mécanisme de pénalisation, certaines zones de l'espace d'exploration des particules sont très pénalisées, y compris la zone de l'objet similaire, vu le mouvement négligeable dans ces zones. La figure 3.7b explique cette deuxième solution et montre bien que la cible reste toujours accessible par le filtre sans risque de divergence. D'où le suivi réussi du système `3VPF` même dans le cas d'une trajectoire irrégulière et avec des variations aigües comme le confirment les images dans le tableau 3.4 et la trajectoire correspondante dans l'image 3.6.

Le tableau 3.4 montre qu'il y a un troisième objet qui bouge parallèlement à la cible. Bien qu'il s'agisse de la main dominée qui a un mouvement moins important



(a) petite taille de la marche aléatoire

(b) grand taille de la marche aléatoire avec pénalisation

FIGURE 3.7 – Le problème de la taille de la marche aléatoire et la solution apportée par le système `3VPF`

que la main dominante, son mouvement accompagne parfois le mouvement de la main dominante avec le même degré d'importance. Nous remarquons, d'après le tableau 3.4, que cette deuxième main ne perturbe pas les systèmes de suivi. Pour les systèmes `PF`, `VPF`et `3VPF` l'estimation est attirée par le visage car quand ils n'arrivent plus à suivre le mouvement rapide de la cible, les filtres ont tendance à se focaliser sur les objets similaires à la cible ayant un mouvement négligeable comme le visage dans ce cas. Nous rappelons ici que pour le système `2VPF`, les coefficients de régularisation de la prise en compte de la vitesse et de l'accélération de l'estimation dans le modèle de mouvement des particules sont des constantes et ne dépendent pas de l'information extraite avec les flots optiques, ils le sont dans le système `3VPF`. Ainsi, le modèle de mouvement des particules du système `2VPF` est moins robuste que le système `3VPF`. En effet, pour le système `3VPF`, avec le mécanisme de pénalisation et les coefficients variables, l'estimation arrive à suivre la cible correctement et reste non attirée par la main dominée (la main gauche). Dans cette situation, c'est le modèle de référence basé sur le vocabulaire de la cible qui renforce cette distinction. La différence d'orientation des mains et les différences fines au niveau des contours semblent être bien prises en compte par les descripteurs de l'observation, principalement les descripteurs de contours, les moments de Zernike et les moments de Hu.

Nous pouvons déduire enfin que nos trois contributions sont bien complémentaires et améliorent considérablement la qualité de suivi même dans les cas les plus difficiles.

Comme mentionné dans la section 2.6 du chapitre précédent, il est difficile de comparer les performances de deux systèmes de suivi sur deux bases de données différentes. À notre connaissance, il y a un seul travail de Yuan et al. qui exploite les bases de données SignStream pour appliquer le suivi de la main [156]. Sur la vidéo $\mathtt{s}_4$, Yuan et al. ont évalué leur système de suivi sur 200 trames uniquement

par rapport à notre évaluation qui s'étale sur la totalité de cette vidéo de 1310 trames. Ils ont conçu leur propre métrique d'évaluation et n'ont pas utilisé aucune des métriques utilisées dans notre travail de suivi, sauf l'évaluation visuelle sur d'autres vidéos de la base. De plus, leur système de suivi est différent des filtres particulaires bien qu'il se base sur le filtrage temporel. Ainsi, il est difficile de se comparer à ce travail. Néanmoins, comme mentionné au début de cette section, le suivi de notre système `3VPF` est considéré comme un « `suivi partiel` » selon la méthode de classement des systèmes de suivi proposée par Bashir et al. (tableau 2.1).

TABLE 3.4 – Le comportement de suivi des quatre systèmes PF, VPF, 2VPF et 3VPF sur un extrait de la vidéo s_4

	Trame #198	Trame #206	Trame #214	Trame #222	Trame #230
PF					
VPF					
2VPF					
3VPF					

Conclusion

Nous avons présenté dans ce chapitre une application de suivi de la main dominante dans des discours de la langue des signes basée sur les filtres particulaires. Nous avons présenté trois contributions pour améliorer la qualité de suivi et nous les avons évaluées séparément pour montrer l'intérêt de chacune de ces contributions. Les résultats de suivi du système complet `3VPF` présentent de meilleurs performances qu'un filtre particuaire classique, en particulier, dans le cas des mouvements rapides et irréguliers de la cible, même en présence d'autres objets similaires qui risquent de perturber le système. Selon la méthode de classement des systèmes de suivi proposée par Bashir et al. [11], le suivi de notre système est considéré comme un « suivi partiel ». De plus, notre système de suivi `3VPF` ne nécessite pas un apprentissage préalable ni de données étiquetées. D'autre part, le système `3VPF` est applicable pour d'autres tâches de suivi, en particulier, le suivi des objets de type déformable.

Néanmoins, il est difficile d'appliquer le système `3VPF` en temps réel à cause de la méthode de détermination du modèle de référence qui nécessite un premier parcours de la vidéo pour la collecte des éléments du vocabulaire de référence. Il serait possible de réaliser le suivi en temps réel en réalisant la construction du vocabulaire simultanément avec le suivi. Le modèle de référence deviendrait alors évolutif avec le suivi. La qualité du suivi peut être encore améliorée par l'application du principe de suivi multiple [1, 46]. Le suivi multiple élimine le risque de confusion lorsque plusieurs objets similaires sont en mouvement. L'application de ce genre de mécanisme peut être une perspective de raffinement du suivi. Il est possible également d'intégrer des règles d'anatomie humaine pour réaliser le suivi des gestes en 3D [77].

Le suivi de la main dominante peut être exploité pour la caractérisation du geste. Le geste dans ce cas est décrit par la trajectoire d'un objet qui contribue principalement à la construction de ce geste et dont le mouvement est dominant. La caractérisation du geste est nécessaire pour l'identifier. Le processus complet de caractérisation du geste et de son identification est connu sous le nom de processus de reconnaissance de gestes. Notre travail de suivi était évalué sur des vidéos de la langue des signes. Cependant, il n'est pas spécifique à la langue des signes. Pour notre prochain travail de reconnaissance des gestes, que nous présentons dans la deuxième partie, nous élargirons le domaine des bases de données pour contenir des gestes appartenant à différents thèmes. Nous détaillons, dans la deuxième partie de ce document, l'état de l'art de la reconnaissance des gestes avec les modèles markoviens et notre application de reconnaissance des gestes avec les contributions proposées.

Deuxième partie

Reconnaissance et détection des gestes

Chapitre 4

État de l'art : Modèles Markovien séquentiels de reconnaissance

Table des matières

4.1 Introduction générale sur les méthodes de reconnaissance dynamique

Les gestes sont des données séquentielles nécessitant des méthodes de reconnaissance dynamiques. En effet, dans le cas de la reconnaissance des discours gestuels, qui sont des ensembles articulés de gestes, la reconnaissance combine deux tâches : la segmentation et la classification. Comme l'affirme Sayre [121], la segmentation et la classification sont deux tâches qui doivent être réalisées simultanément. La tâche de segmentation doit déterminer les limites des gestes sur la séquence. La tâche de classification doit attribuer à chaque sous-séquence une étiquette appartenant à un vocabulaire donné. La tâche de classification doit intégrer également des connaissances *a priori* sur les données telles que le vocabulaire des gestes, les durées des gestes, l'environnement d'enregistrement, etc. La difficulté principale que comporte l'étape de segmentation est la variabilité de la durée des instances d'un même geste. L'étape de classification est confrontée à la variabilité des instances d'un même geste, à laquelle elle doit être robuste également.

Il existe deux catégories de méthodes de reconnaissance dynamiques, que nous pouvons appeler également « *classifieurs* » ; les méthodes de reconnaissance sans apprentissage et les méthodes de reconnaissance avec apprentissage. Les méthodes de reconnaissance sans apprentissage sont les premières méthodes de classification proposées dans la littérature. Elles sont basées sur l'appariement comme la méthode de déformation temporelle dynamique, appelé également « Dynamic Time Warping » [14]. Leur principe repose sur la comparaison à la séquence à identifier à l'aide un représentant, appelé également « *template* », de chaque classe du problème en utilisant une distance « élastique » capable d'absorber dans une certaine mesure, la variabilité de la durée. À l'opposé, les méthodes de reconnaissance dynamiques avec apprentissage, principalement les modèles markoviens, optimisent pendant l'étape d'apprentissage des modèles graphiques statistiques représentant le vocabulaire étiqueté. La reconnaissance avec ces méthodes est réalisée en alignant ces modèles graphiques sur chaque donnée de test et en utilisant également un algorithme d'appariement élastique, dont l'algorithme de `Viterbi` [140] est une instance.

L'avantage des méthodes de reconnaissance dynamiques avec apprentissage est la génération de ces modèles graphiques probabilistes intégrant l'information portée par les exemples d'apprentissage et qui permet d'optimiser le modèle des classes pour qu'il représente le mieux possible les données réelles. Les modèles markoviens ont la capacité d'intégrer des connaissances *a priori* telles que les règles grammaticales des données linguistiques. Ces connaissances *a priori* facilitent la reconnaissance et diminuent le risque d'erreur causée par le manque de distinction descriptive.

Ces avantages, maintenant bien mis en évidence dans la littérature, justifient

notre intérêt pour les modèles statistiques en général afin de réaliser la tâche de reconnaissance de gestes. Nous nous intéressons tout d'abord aux modèles de Markov cachés (les MMC) qui sont des modèles markoviens à l'état de l'art largement utilisés pour la reconnaissance de données séquentielles [109]. Afin de modéliser explicitement la durée de ces données séquentielles, une variante semi-markovienne des MMC est proposée [39]. Nous allons expliquer au début de ce chapitre, dans la section 4.2, les principes des MMC et leur variante, les semi-MMC. Nous donnons ensuite un aperçu de la littérature sur les applications des MMC à la reconnaissance de gestes.

Comme nous l'avons mentionné au début de cette introduction, la tâche de reconnaissance combine deux tâches : la segmentation et la classification. Les MMC sont adaptés au problème de segmentation [109]. Il existe d'autre modèles markoviens de séquences, les champs aléatoires conditionnels (les CAC) [143], qui sont adaptés au problème de classification [57]. Nous nous intéressons à ces modèles qui représentent un modèle markovien récemment développé [74]. Afin de modéliser explicitement la durée des données séquentielles comme dans le cas des MMC, une version semi-markovienne est développée pour les CRF [119]. D'autre part, afin d'adapter les CAC au problème de la modélisation de la structure globale des données et au problème de la segmentation, les CAC cachés [108] et les CAC dynamiques latents [91] ont été proposés. Nous détaillons le principe et les avantages des CAC dans la deuxième partie de ce chapitre, dans la section 4.3, et présentons trois variantes ; les CAC semi-Markoviens, les CAC cachés et les CAC dynamiques latents. Nous donnerons également un aperçu sur les applications des CAC pour la reconnaissance de gestes dans la littérature. À la fin de ce chapitre, dans la section 4.4, nous donnerons un aperçu sur les modèles hybrides, proposés dans la littérature, qui combinent les MMC à des méthodes de classification.

4.2 Les modèles de Markov à états cachés : les MMC

Les MMC sont des modèles statistiques qui permettent de réaliser une reconnaissance dynamique. Ils sont bien connus pour leur capacité à segmenter les données séquentielles. Nous présentons dans cette section la théorie de ces modèles et leur variante les modèles semi-Markoviens à états cachés. Cette variante modélise explicitement la durée des données séquentielles. Nous présentons à la fin de cette section un aperçu sur les applications des MMC pour la reconnaissance des gestes et nous indiquerons les avantages et les limites de cette méthode.

4.2.1 Théorie

Les MMC [12] sont des modèles probabilistes génératifs. Leur principe consiste à générer des observations en se basant sur des états cachés. Ils calculent une probabilité jointe $p(y_{1:T}, x_{1:T})$ (équation 4.1) sur les observations $x_{1:T}$ et les états

cachés $y_{1:T}$. La forme de l'équation 4.1 de cette probabilité est obtenue en appliquant deux hypothèses : chaque observation x_t ne dépend que de l'état caché y_t (indépendance des observations conditionnées sur les états) et chaque état caché y_t ne dépend que de l'état qui le précède y_{t-1} (hypothèse Markovienne à l'ordre 1). La figure 4.1 présente graphiquement un `MMC` au cours du temps et montre les dépendances fondées sur ces deux hypothèses, qui conduit à la factorisation de l'équation 4.1.

$$p(y_{1:T}, x_{1:T}) = p(y_1)p(x_1|y_1)\prod_{t=2}^{T} p(y_t|y_{t-1})p(x_t|y_t) \tag{4.1}$$

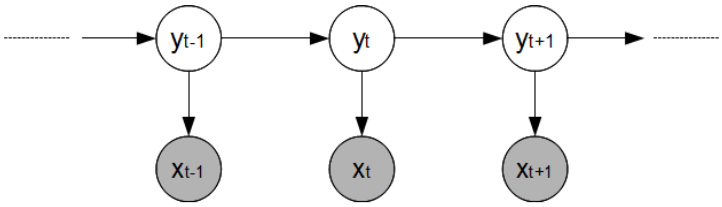


FIGURE 4.1 – La représentation graphique d'un `MMC` : chaque observation x_t ne dépend que de l'état caché y_t et chaque état caché y_t ne dépend que de l'état qui le précède y_{t-1}

Soient l'ensemble des états $\mathcal{S} = \{s_1, s_2, .., s_{N_s}\}$ et l'ensemble des observations $\mathcal{O} = \{o_1, o_2, .., o_{N_o}\}$. y_t est une variable aléatoire qui prend des valeurs dans $\mathcal{S}$ et x_t est une variable aléatoire qui prend des valeurs dans $\mathcal{O}$. Un `MMC` est défini par le quintuplé $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{B}, \pi)$. $\mathcal{A}$ définit une matrice de probabilités de transition entre les états cachés y_t comme suit :

$$\begin{array}{rccl} \mathcal{A}: & \mathcal{S}\times\mathcal{S} & \longrightarrow & [0,1] \\ & (s_i, s_j) & \longmapsto & p(y_t = s_i | y_{t-1} = s_j) \end{array} \tag{4.2}$$

Les `MMC` discrets modélisent les caractéristiques discrètes à travers une matrice de probabilités d'émission notée $\mathcal{B}$ calculée directement lors de l'apprentissage. $\mathcal{B}$ définit les probabilités d'émission des observations x_t à partir des états y_t comme suit :

$$\begin{array}{rccl} \mathcal{B}: & \mathcal{O}\times\mathcal{S} & \longrightarrow & [0,1] \\ & (o_k, s_i) & \longmapsto & p(x_t = o_k | y_t = s_i) \end{array} \tag{4.3}$$

Dans le cas des données continues, les probabilités de génération sont représentées par des densités de probabilité conditionnelles modélisées chacune par un mélange de G Gaussiennes, appelé également « *Gaussian Mixture Model* », comme suit : $p(x_t = o_k | y_t = s_i) = \sum_{g=1}^{G} w_{ig} \times \mathcal{N}(o_k; \mu_{ig}, \Sigma_{ig})$. Chaque gaussienne, correspondant à un état s_i, est définie par une moyenne μ_{ig} et une matrice

de covariance Σ_{ig}, qui est en générale une matrice diagonale. w_{ig} représente la probabilité *a priori* de la gaussienne g correspondant à l'état s_i. Cette modélisation gaussiennes des données réelles est spécifique aux `MMC` continus.

Enfin, π définit les probabilités des états initiaux comme suit :

$$\begin{array}{rccc} \pi : & \mathcal{S} & \longrightarrow & [0,1] \\ & s_i & \longmapsto & p(y_1 = s_i) \end{array} \tag{4.4}$$

L'apprentissage d'un `MMC` nécessite l'apprentissage des paramètres $\Theta =< \mathcal{A}, \mathcal{B}, \pi >$. Quand l'étiquetage des données d'apprentissage est incomplet, autrement dit lorsque les trames ne sont pas étiquetées, l'apprentissage des `MMC` permet d'estimer des séquences d'étiquettes correspondantes grâce à l'apprentissage de `Baum-Welch`. L'algorithme de `Baum-Welch` [147] permet de réaliser cet apprentissage en tenant compte de toute les séquences d'états possibles. Cet algorithme permet de déterminer les probabilités des états initiaux, les probabilités de transition entre états et les paramètres des probabilités de génération des observations à travers la maximisation de la vraisemblance V définie selon l'équation 4.5.

$$V(\Theta) = \prod_{n=1}^{N_{seq}} p(X^n|\Theta) \tag{4.5}$$

où $X^n = (x_1^n, x_2^n, .., x_t^n, .., x_{T_n}^n)\ \forall n \in [1, N_{seq}]$ sont les séquences d'observation des données d'apprentissage et Θ représente les paramètres du `MMC`. Θ est mis à jour à chaque itération d'apprentissage en appliquant l'algorithme Espérance-Maximisation (EM) [37] mis en œuvre par l'algorithme `Baum-Welch`. Pour simplifier les calculs, il est possible d'utiliser le log-vraisemblance comme suit :

$$LV(\Theta) = \sum_{n=1}^{N_{seq}} log(p(X^n|\Theta)) \tag{4.6}$$

La phase d'inférence permet de déterminer la séquence d'états cachés Y^* la plus probable qui décrit la séquence d'observations X donnée en entrée. L'algorithme de `Viterbi` [140] ou l'algorithme `forward-backward` [110] permettent de trouver cette meilleur séquence selon deux critères différents.

L'algorithme de `Viterbi` permet de déterminer globalement la meilleure séquence d'états cachés $Y^* = \arg\max_Y(X^n, Y)$ en calculant une vraisemblance récursive $\delta_{t,i}$ (équation 4.7) et en appliquant à la fin de la séquence de trames un processus de *retour sur trace*, appelé également le processus de *backtracking*.

$$\delta_{t,i} = p(x_t|y_t = s_i) \max_{s_j \in \mathcal{S}}(p(y_t = s_i|y_{t-1} = s_j)\delta_{t-1,j}) \tag{4.7}$$

avec

$$\delta_{1,i} = p(x_1|y_1 = s_i)\pi_i$$

L'algorithme de Viterbi peut être appliqué pour l'apprentissage des `MMC` selon un processus itératif en estimant la meilleur séquence d'états cachés sur les données d'apprentissage et en maximisant la vraisemblance $V(\Theta)$ [3]. L'algorithme de Viterbi est souvent appliqué pour le décodage des données de test en cherchant la meilleur séquence d'étiquettes pour chaque donnée de test.

L'algorithme `forward-backward` détermine la probabilité de chaque état pour chaque trame en tenant en compte des observations passées et des observations futures. Cet algorithme nécessite trois parcours de la séquence de trames : un parcours pour calculer les probabilités forward selon l'équation 4.8, un parcours pour calculer les probabilités backward selon l'équation 4.9 et un parcours pour combiner les deux types de probabilités selon l'équation 4.10 et déduire par la suite la probabilité de chaque état pour chaque trame. La meilleur séquences d'états cachés Y^* dans ce cas peut être déterminée par la séquence d'états ayant les probabilités locales γ maximales.

$$\alpha_t(i) = p(x_1 x_2 ..x_t, y_t = s_i) = \sum_{j=1}^{N_s} \alpha_{t-1}(j) p(y_t = s_i | y_{t-1} = s_j) p(x_t | y_t = s_i) \quad (4.8)$$

avec

$$\alpha_1(i) = p(x_1 | y_1 = s_i)\pi_i$$

$$\beta_t(i) = p(x_{t+1} x_{t+2} ..x_T, y_t = s_i) = \sum_{j=1}^{N_s} \beta_{t+1}(j) p(y_{t+1} = s_j | y_t = s_i) p(x_t | y_t = s_i) \quad (4.9)$$

$$\gamma_t(i) = p(y_t = i | X) = \frac{p(y_t = i, X)}{p(X)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N_s} \alpha_t(j)\beta_t(j)} \quad (4.10)$$

Outre les `MMC`, l'algorithme forward-backward et l'algorithme de Viterbi sont également appliqués pour l'apprentissage et le décodage des données de tests avec d'autre modèles Markoviens tel que les `CAC`.

Selon Kasteren [136], la principale limite des `MMC` est le manque de flexibilité de la modélisation de la durée au niveau des états. En effet, La probabilité de séjourner dans un état i pour une durée d est une probabilité géométrique de la forme $p_i(d) = (a_{ii})^{d-1}(1 - a_{ii})$ où a_{ii} est la probabilité d'auto-transition de l'état i. Cette forme géométrique découle directement de la définition du modèle `MMC` et il n'est pas possible de la modéliser autrement dans ce cas. Or, la distribution géométrique ne peut pas représenter toutes les formes de distribution de la durée. La variante semi-markovienne des `MMC` fournit la possibilité de modéliser explicitement la durée de séjour dans les états. Nous allons présenter cette variante dans la sous-section suivante.

4.2.2 Les semi-MMC

Les semi-MMC [39] ont la structure des MMC mais le processus des états cachés est semi-Markovien plutôt que Markovien. Le processus Markovien suppose que la prise de décision locale selon les probabilités de transition est instantanée. Le processus semi-markovien suppose qu'il existe une durée de séjour dans chaque état caché notée d_u. Ainsi, les états cachés deviennent des méta-états η_u constitués d'une étiquette y_u et d'une durée de séjour d_u comme le montre la figure 4.2. Cette durée permet au modèle semi-MMC d'associer une étiquette à plusieurs observations séquentielles.

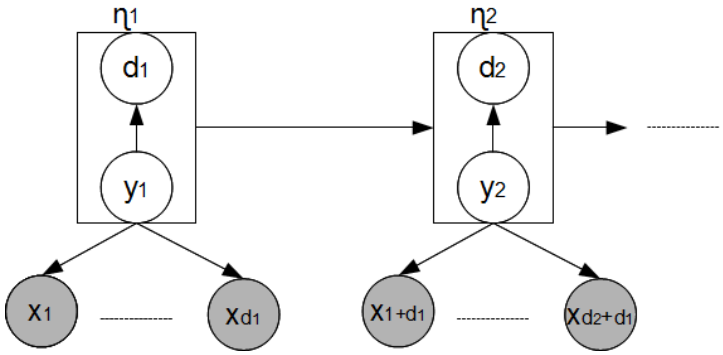


FIGURE 4.2 – La représentation graphique du système Semi-MMC

La forme de la probabilité jointe $p(\eta_{1:U}, x_{1:T}) = p(< d_{1:U}, y_{1:U} >, x_{1:T})$ sur les observations $x_{1:T}$ et les méta-états cachés $\eta_{1:U}$, telle que $U < T = \sum_{u=1}^{U} d_u$, est donnée par l'équation 4.11. Cette probabilité suppose une segmentation de la séquence d'observations selon les méta-états au rythme des durées $d_{1:U}$.

$$p(\eta_{1:U}, x_{1:T}) = P_I \times P_S \times P_O \tag{4.11}$$

où

$$P_I = p(y_1)p(d_1|y_1)\prod_{t=1}^{d_1} p(x_t|y_1)$$

$$P_S = \prod_{u=2}^{U} p(y_u|y_{u-1})p(d_u|y_u)$$

$$P_O = \prod_{u=2}^{U}\left[\prod_{(t=\sum_{k=1}^{u-1} d_k)}^{(\sum_{k=1}^{u} d_k)} p(x_t|y_u)\right]$$

Le terme P_I représente la probabilité du méta-état caché initial. Le terme P_S représente la probabilité de la segmentation de la séquence. Il est le produit des probabilités de transition entre les méta-états cachés et les probabilités des

durées. Le terme P_O représente le produit des probabilités de génération d'une séquence d'observations à partir des méta-états cachés.

L'apprentissage d'un semi-MMC est similaire à celui d'un MMC, l'état caché est remplacé par le méta-état caché. Les probabilités $p(d_u|y_u)$ peuvent être déterminées avec des connaissances *a priori* [136].

La meilleure séquence de méta-états cachés peut être déterminée avec une version adaptée de l'algorithme de Viterbi.

Les semi-MMC fournissent la possibilité d'une modélisation explicite et flexible de la durée des données séquentielles, mais la complexité de leur procédure d'apprentissage est élevée dans le cas où les durées des données d'apprentissage sont non connues. La procédure d'apprentissage nécessite alors de considérer toutes les durées possibles à chaque étape. Quand les données sont totalement étiquetées, la procédure d'apprentissage ne nécessite pas d'étape d'inférence [136].

Dans la prochaine sous-section, nous allons donner un aperçu de la littérature traitant des applications des MMC à la reconnaissance de gestes.

4.2.3 Applications

Les MMC et leurs variantes sont exploités dans les domaines du traitement du signal et de la modélisation de séquences. Ils sont largement utilisés pour la reconnaissance de la parole [109], la reconnaissance de l'écriture [90], la modélisation des séquences biologiques [95]... Ils sont également appliqués dans le domaine de la reconnaissance de gestes [141, 142, 101, 136, 5, 102, 130, 120, 52, 8].

Une large communauté de chercheurs étudiant la reconnaissance de gestes se spécialise dans l'étude de la langue des signes. Par exemple Vogler et al. [141], Agris et al. [142] et Ong et al. [101] ont conçu un modèle de MMC parallèles pour la reconnaissance des phrases signées. Ils ont pour cela distingué des descripteurs des gestes, tels que la position, l'orientation et la distance comme expliqué dans la figure 4.3, afin de faciliter le processus d'apprentissage des MMC et optimiser l'exploitation de ces descripteurs. Cette décomposition se manifeste par la génération d'un MMC pour chaque type de descripteur et pour chaque unité élémentaire (une sous-unité du signe).

Les MMC ont été également exploités pour la réalisation d'un système de reconnaissance avec un nombre très réduit d'exemples d'apprentissage [73, 62, 146, 153]. Cela permet de remédier au problème de manque de données qui est un problème capital dans le domaine de l'apprentissage artificiel. Konecny [73] et al., Jackson [62] et Weiss [146] ont proposé un modèle MMC global de reconnaissance de séquences de gestes (figure 4.4) en utilisant des bases d'apprentissage mono-exemple. Ce modèle global est sous la forme d'un ensemble de MMC « gauche droite » interconnectés modélisant chacun un geste. De chaque état de chaque MMC, il est possible de transiter vers l'état lui-même ou vers un état postérieur interne ou externe. Dans la version proposée par Jackson [62], chaque trame du geste est modélisée

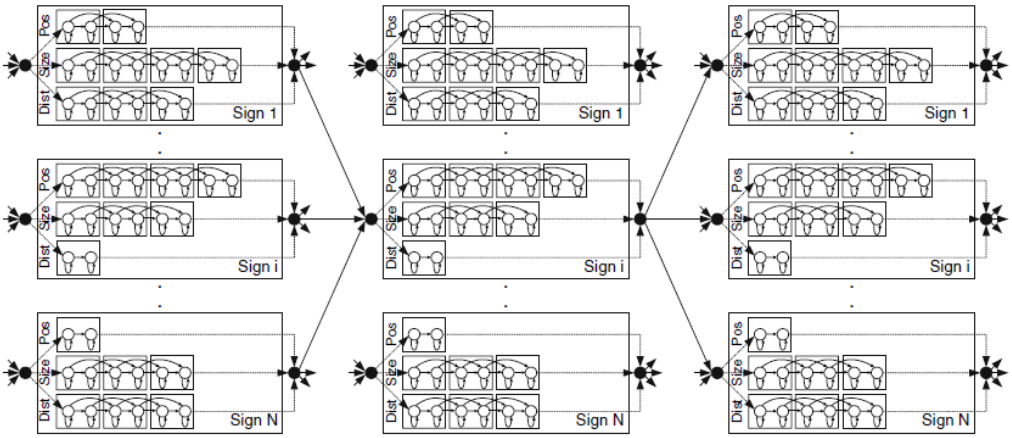


FIGURE 4.3 – Un modèle de MMC parallèles pour la reconnaissance des phrases en langue des signes selon Kraiss et al. [142]

par un état. La figure 4.4 montre que ce modèle est complexe à cause des sauts et des connexions complètes.

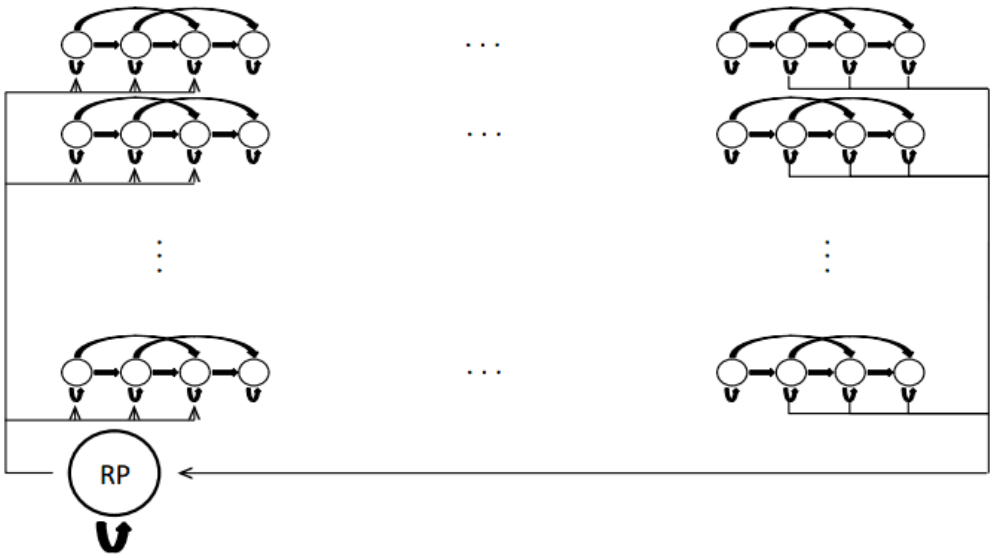


FIGURE 4.4 – Le modèle MMC global pour la reconnaissance des séquences de gestes généré avec un seul exemple d'apprentissage par classe selon Konecny et al. [73]

En ce qui concerne la reconnaissance de l'activité humaine, nous pouvons mentionner le travail de Kasteren et al. [136] qui a étudié la variante semi-markovienne des MMC. Comme mentionné auparavant, les MMC sont limités au niveau de la modélisation du temps de séjour dans les états du modèle. Kasteren a montré qu'il est possible de résoudre ce problème avec les modèles semi-markoviens [136]. Il a montré que ces modèles atteignent des performances de reconnaissance dépassant les

performances des modèles markoviens. Néanmoins, les modèles semi-markoviens présentent une certaine complexité de calcul. Dans le cas où la durée des éléments des données n'est pas fournie dans la vérité terrain, il est nécessaire que le système de reconnaissance effectue une étape d'inférence pour tester toutes les durées possibles pendant l'étape d'apprentissage.

4.2.4 Avantages et inconvénients des MMC

La modélisation graphique des données avec un modèle MMC est très intéressante. Ce modèle permet de guider le système de décodage et de préserver une certaine continuité et cohérence structurelle des données. Ce modèle permet en effet d'intégrer des connaissances *a priori* de haut niveau telles que les règles grammaticales des données linguistiques. Un autre avantage des MMC est qu'ils ne nécessitent pas un étiquetage local des trames des données d'apprentissage grâce à leur processus d'apprentissage de type EM.

D'autre part, les modèles génératifs comme les MMC modèlisent les caractéristiques des données avec des distributions gaussiennes. Dans le cas de manque de données d'apprentissage, la modélisation devient pauvre et inadéquate ce qui présente un inconvénient principal des MMC. Cependant, les modèles discriminants peuvent remédier à ce problème. Nous présentons dans la section suivante un modèle séquentiels markovien discriminant : les CAC. Ce modèle a été proposé par Lafferty et al. [74] en 2001. Il présente certains avantages qui peuvent remédier aux problèmes des MMC. Il présente donc des propriétés complémentaires à celles d'un MMC. Ces avantages seront détaillés dans la sous-section 4.3.6.

4.3 Les champs aléatoires conditionnels : les CAC

Les Champs Aléatoires Conditionnels (CAC) sont des modèles markoviens discriminants connus pour leur capacité de classification. Nous allons présenter dans cette section la théorie des CAC et un ensemble de leurs variantes : les semi-Markov CAC, les CAC cachés et les CAC dynamiques latents. Les semi-Markov CAC modélisent explicitement la durée de séjour dans les états du modèles comme dans le cas des semi-MMC. Les CAC cachés réalisent la cohérence structurelle des données reconnues et les CAC dynamiques latents répondent au problème de segmentation.

4.3.1 Théorie

Les CAC sont des modèles probabilistes discriminants basés sur une approche conditionnelle. Ils servent à étiqueter les trames des séquences de données. À chaque instant, ils ont une capacité à prendre en compte tout le contexte de l'observation et d'analyser toutes les trames de cette observation. Les CAC fusionnent les caractéristiques des observations en appliquant le principe d'activation et de

désactivation des poids des fonctions caractéristiques, appelées également fonctions potentielles. La représentation graphique d'un CAC est un graphe non orienté linéaire ayant une structure proche de celle d'un MMC. La différence entre les deux modèles est qu'avec les CAC l'hypothèse d'indépendance conditionnelle des observations n'est plus nécessaire, ce qui se matérialise par le fait que chaque état peut dépendre de toute la séquence d'observations X ou d'une partie de celle-ci, d'où les liaisons mutuelles reliant les états et la séquence d'observations comme le montre la figure 4.5.

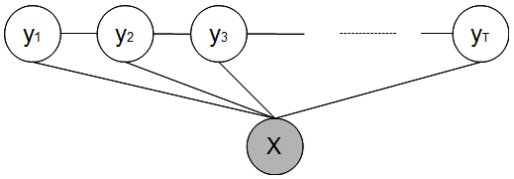


FIGURE 4.5 – Une représentation de la structure graphique des CAC linéaires

La probabilité de la séquence des états $Y = y_{1:T}$ sachant la séquence d'observations $X = x_{1:T}$ est calculée par :

$$p(Y|X) = \frac{1}{Z(X)} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(y_{t-1}, y_t, X, t)\right) \tag{4.12}$$

où $Z(X)$ est un terme de normalisation défini par :

$$Z(X) = \sum_{s_i, s_j \in \mathcal{S}} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(y_{t-1} = s_j, y_t = s_i, X, t)\right) \tag{4.13}$$

$f_k, \forall k \in [1, K]$ représente les fonctions caractéristiques. Il existe deux types de fonctions caractéristiques : les fonctions caractéristiques de transitions entre états successifs représentant la dépendance Markovienne (équation 4.14) où k est associé à un couple de valeurs d'états $(s_i, s_j) \in \mathcal{S}^2$, et les fonctions caractéristiques d'observations (équation 4.15) où k est associé à un couple état-observation (s_i, o_{s_i}). λ_k est un poids de la fonction f_k. Les poids λ_k $\forall k \in [1, K]$ sont les paramètres optimisés durant la procédure d'apprentissage des CAC.

$$f_{(s_j,s_i)}(y_{t-1}, y_t) = \begin{cases} 1 & \text{si } y_{t-1} = s_j \text{ et } y_t = s_i \\ 0 & \text{sinon} \end{cases} \tag{4.14}$$

$$f_{(s_i,o_{s_i})}(y_t, X) = \begin{cases} o_{s_i}(X) & \text{si } y_t = s_i \\ 0 & \text{sinon} \end{cases} \tag{4.15}$$

Dans le cas où les caractéristiques des observations sont réelles, $o_{s_i}(X)$ représente une combinaison de valeurs réelles correspondant à un ensemble de ces

caractéristiques, on parle dans ce cas de CAC continu. Dans le cas où les caractéristiques des observations sont discrètes, $o_{s_i}(X)$ représente une valeur binaire activée ou désactivée selon une condition sur un ensemble de ces caractéristiques, on parle dans ce cas de CAC discret. Les fonctions $o_{s_i}(.)$ $\forall s_i \in \mathcal{S}$ doivent être définies par l'utilisateur.

La procédure d'apprentissage de ces modèles probabilistes agit sur les poids λ_k $\forall k \in [1, K]$, que nous notons $\lambda = (\lambda_1, \lambda_2, .., \lambda_K)$. Le critère d'apprentissage est la maximisation du log-vraisemblance LV :

$$LV(\lambda) = \sum_{n=1}^{N_{seq}} \log p(Y^n | X^n, \lambda) \tag{4.16}$$

$$LV(\lambda) = \sum_{n=1}^{N_{seq}} \left(\sum_{t=1}^{T} \sum_{k=1}^{K} \lambda_k f_k(y_{t-1}^n, y_t^n, X^n, t) - \log Z(X^n, \lambda) \right) \tag{4.17}$$

où N_{seq} est le nombre de séquences de trames de la base d'apprentissage et Y^n $\forall n \in [1, N_{seq}]$ représentent la vérité terrain au niveau trames des séquences d'apprentissage. Il n'y pas d'apprentissage de type EM comme dans le cas des MMC. Ainsi, les CAC nécessitent un étiquetage trame par trame de la base d'apprentissage. Tout algorithme de descente de gradient peut être utilisé pour calculer l'optimum de la fonction $LV(\lambda)$ tel que l'algorithme BFGS, sa version à mémoire réduite L-BFGS [82] et l'algorithme SGD [1] (Stochastic Gradient Descent).

Une version adaptée de l'algorithme de Viterbi ou de l'algorithme forward-backward [109] peut être utilisée pour déterminer la meilleure séquence d'étiquettes pour les données de test.

Les CAC sont principalement des méthodes de classification. L'étape d'apprentissage génèrent un modèle CAC pour toutes les classes de données de la base d'apprentissage. Ainsi, il est possible que les CAC attribuent à des trames successives des étiquettes d'une même classe en introduisant quelques étiquettes appartenant à d'autres classes. Autrement dit, la notion de continuité et de cohérence structurelle n'est pas prise en compte par les CAC. En conséquence, la notion de durée n'est pas modélisée par les CAC. Pour prendre en compte cette notion de durée comme dans le cas des semi-MMC [119], les semi-Markov CAC ont été proposés. Dans la sous-section suivante, nous allons donner un aperçu de cette variante des CAC.

4.3.2 Les semi-Markov CAC

Les semi-Markov CAC [119] ont la capacité de modéliser les durées des états comme dans le cas des semi-MMC. En effet, les étiquettes sont remplacées par des segments $\eta_{1:U} =< d_{1:U}, y_{1:U} >$, comme montré sur la figure 4.6, où d_u est la durée

1. http ://en.wikipedia.org/wiki/Stochastic_gradient_descent

de séjour dans l'état y_u. Le vecteur $\eta_{1:U}$ présente une segmentation de la séquence d'observation $X = x_{1:T}$. Selon cette définition, les semi-Markov CAC ont une capacité supplémentaire de segmentation des séquences d'observations. L'équation de probabilité *a posteriori* des semi-CAC est donnée par l'équation 4.18. La différence entre cette forme de probabilité et la forme classique est la prise en compte de la durée d_u par les fonctions caractéristiques f_k.

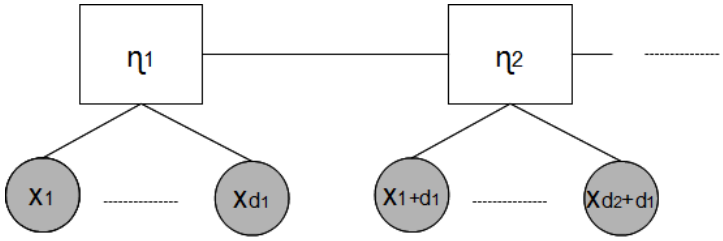


FIGURE 4.6 – La représentation graphique du modèle semi-CAC

$$p(\eta_{1:U}|X) = \frac{1}{Z(X)} \exp\left(\sum_{u=1}^{U}\sum_{k=1}^{K} \lambda_k f_k(y_{u-1}, y_u, X, u, d_u)\right) \tag{4.18}$$

où

$$Z(X) = \sum_{s_i, s_j \in \mathcal{S}} \exp\left(\sum_{u=1}^{U}\sum_{k=1}^{K} \lambda_k f_k(y_{u-1} = s_j, y_u = s_i, X, u, d_u)\right) \tag{4.19}$$

Les mêmes méthodes d'apprentissage et d'inférence utilisées pour les CAC sont applicables pour les semi-CAC dans le cas où l'on dispose des données étiquetées au niveau trames.

Comme dans le cas des semi-MMC, la complexité de la procédure d'apprentissage des semi-CAC est très élevée lorsqu'on n'a pas l'étiquetage au niveau trames. En premier lieu, cela est dû à la durée non connue des segments dans une nouvelle séquence d'observation. La procédure d'apprentissage nécessite ainsi de considérer toutes les durées possibles à chaque étape. En deuxième lieu, cette complexité élevée est due à la nécessité d'une étape d'inférence à chaque itération de l'apprentissage [136].

En général, les CAC ont un problème de modélisation des caractéristiques continues. Nous montrerons dans le chapitre 5 l'écart important entre les performances de reconnaissance des CAC continus et des CAC discrets en quantifiant les valeurs réelles des caractéristiques. Quattoni et al. ont alors ajouté une couche d'états cachés aux CAC claissiques pour remédier à ce problème [108]. Dans la sous-section suivante, nous allons présenter cette variante des CAC, nommée les CAC cachés.

4.3.3 Les CAC cachés

Les CAC cachés [108] permettent d'introduire un niveau de description supplémentaire de la séquence d'observations $X = x_{1:T}$ à travers une couche d'états cachés $H = h_{1:T}$ comme le montre la figure 4.7. Cette représentation explicite des caractéristiques facilite la prise en compte des caractéristiques continues. Les états cachés permettent de combiner les caractéristiques continues pour constituer des méta-caractéristiques à l'image des caractéristiques discrètes des CAC discrets. Une équivalence peut également être établie entre les CAC cachés et les MMC continus [53].

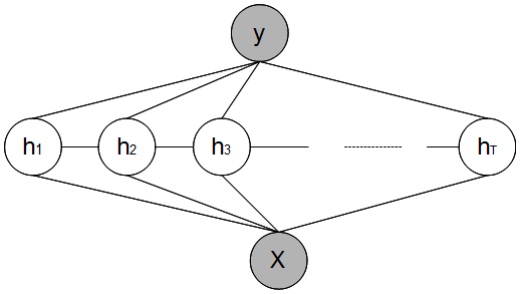


FIGURE 4.7 – Une représentation graphique des CAC cachés

La procédure de classification des données séquentielles avec les CAC cachés repose sur l'attribution d'une seule étiquette y à chaque donnée. Ainsi, les CAC cachés ont la capacité de partitionner implicitement une étiquette y en sous-unités $h_{1:T}$. Les états cachés h_t prennent des valeurs τ_l appartenant un ensemble d'états cachés $\mathcal{E}$. La probabilité *a posteriori* des CAC cachés s'écrit comme suit :

$$p(y|X) = \sum_{\tau_{1:T} \in \mathcal{E}^T} p(y, h_{1:T} = \tau_{1:T}|X) \tag{4.20}$$

tel que

$$p(y, h_{1:T}|X) = \frac{1}{Z(X)} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(h_{t-1}, h_t, y, X, t)\right) \tag{4.21}$$

où

$$Z(X) = \sum_{\tau_l, \tau_m \in \mathcal{E}, s_i \in \mathcal{S}} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(h_{t-1} = \tau_l, h_t = \tau_m, y = s_i, X, t)\right) \tag{4.22}$$

Les fonctions caractéristiques f_k représentent les liaisons de transitions entre deux états cachés successifs h_{t-1} et h_t, les liaisons d'observations entre un état caché h_t et la donnée X, et les liaisons entre l'étiquette y et les états cachés $h_{1:T}$.

En passant au log de vraisemblance, il est possible de réaliser l'apprentissage des CAC cachés avec la méthode de gradient conjugué et la méthode de propagation de croyance comme proposé dans [108]. La procédure d'apprentissage peut être également réalisée avec une technique d'optimisation Quasi-Newton comme le L-BFGS [145].

L'algorithme de Viterbi peut également être adapté pour décoder les données de test avec les CAC cachés.

Les CAC cachés dans leur version de base n'ont pas la capacité de segmenter la séquence en plusieurs classes. Ainsi, les CAC dynamiques latents ont été conçus pour répondre à ce problème [91]. Nous présentons cette variante des CAC dans la sous-section suivante.

4.3.4 Les CAC dynamiques latents : les CACDL

L'apprentissage d'un CAC classique nécessite un étiquetage trame par trame des séquences d'apprentissage. En d'autre termes, les CAC apprennent à reconnaitre mais il faut leur fournir la segmentation exacte des données d'apprentissage. Les CAC dynamiques latents sont des CAC que l'on peut apprendre sans leur fournir la segmentation exacte de la séquence d'observations.

Comme le montre la figure 4.8, les CACDL [91] sont une extension des CAC cachés. Les CACDL n'attribuent pas une seule étiquette à toute la donnée séquentielle comme dans le cas des CAC cachés mais une étiquette pour chaque trame de la donnée. Ces étiquettes sont représentées par le vecteur $Y = y_{1:T}$. Comme chaque étiquette y_t est liée à tous les états cachés $h_{1:T} = H$, la segmentation de l'observation $X = x_{1:T}$ devient possible avec les CACDL.

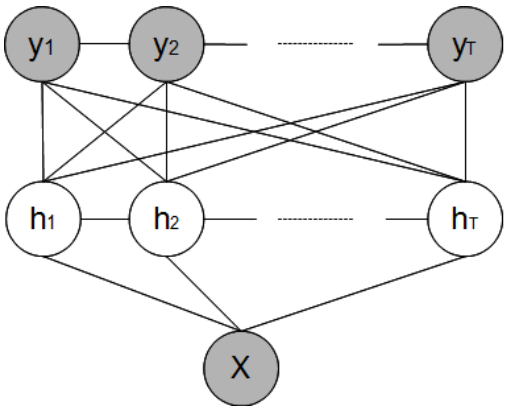


FIGURE 4.8 – Une représentation graphique des CACDL

Ainsi, les CACDL ont la capacité de segmenter et d'étiqueter simultanément les trames observées lors de l'apprentissage. Ces capacités permettent de capturer d'une part la dynamique extrinsèque du modèle à travers la modélisation d'un flux continu des classes des étiquettes, et d'autre part la dynamique intrinsèque

du modèle à travers la modélisation des sous-unités des classes des séquences d'observations en intégrant des états cachés. La probabilité *a posteriori* calculée par les `CACDL` est donnée par :

$$p(y_{1:T}|X) = \sum_{\tau_{1:T} \in \mathcal{E}^T} p(y_{1:T}|h_{1:T}, X) p(h_{1:T}|X) \tag{4.23}$$

or les y_t ne dépendant pas directement de X, ainsi $p(y_{1:T}|h_{1:T}, X) = p(y_{1:T}|H)$.

En appliquant la forme classique de la probabilité *a posteriori* des `CAC`, nous obtenons :

$$p(y_{1:T}|H) = \frac{1}{Z(H)} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K'} \lambda'_k f'_k(y_{t-1}, y_t, y_{t-1}, H, t)\right) \tag{4.24}$$

où

$$Z(H) = \sum_{s_j, s_i \in \mathcal{S}} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K'} \lambda'_k f'_k(y_{t-1} = s_j, y_t = s_i, H, t)\right) \tag{4.25}$$

et

$$p(h_{1:T}|X) = \frac{1}{Z(X)} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(h_{t-1}, h_t, X, t)\right) \tag{4.26}$$

où

$$Z(X) = \sum_{\tau_l, \tau_m \in \mathcal{E}} \exp\left(\sum_{t=1}^{T}\sum_{k=1}^{K} \lambda_k f_k(h_{t-1} = s_l, h_t = s_m, X, t)\right) \tag{4.27}$$

La procédure d'apprentissage de ce modèle est similaire à la procédure d'apprentissage du modèle `CAC` cachés. L'apprentissage peut être réalisé avec les probabilités marginales et la propagation de croyances, et avec la méthode d'optimisation Quasi-Newton `BFGS`.

Il y a deux algorithmes possibles pour chercher la meilleure séquence d'étiquettes qui labélise les données : l'algorithme de `Viterbi` et l'approche des probabilités marginales maximales calculées par la méthode de propagation de croyances [91].

Dans la prochaine sous-section, nous allons donner une synthèse sur les variantes des `CAC` et un aperçu des applications des `CAC` à la reconnaissance de gestes proposées dans la littérature.

4.3.5 Synthèse et Applications

Les CAC sont une méthode de classification qui a été appliquée dans différents domaines. Feng et al. [38] l'ont appliqué pour reconnaitre les documents historiques manuscrits. Hébert et al. [57] ont appliqué également les CAC pour l'extraction de structures dans les images de documents. Cependant, dans la plupart des travaux sur les données séquentielles, notamment les travaux de reconnaissance de la langue des signes, les chercheurs ont appliqué des variantes augmentées des CAC [145, 91, 150, 151, 28, 91]. En effet, les CAC, avec leur version standard, ne sont pas capables de reconnaitre des données séquentielles conformément à un modèle structurel, tel qu'un modèle de langage. Autrement dit, ils étiquettent trame par trame les données séquentielles sans pouvoir les regrouper selon un modèle de haut niveau tel qu'un modèle de langage. D'où le recours à des variantes des CAC ajoutant cette capacité. Les variantes qui sont fréquemment utilisées pour la reconnaissance des données séquentielles sont les semi-Markov CAC [136] et les CAC cachés [145]. Les CAC cachés identifient une séquence complète par une seule étiquette. Ainsi, leur exploitation se limite à la reconnaissance des gestes isolés. Les semi-Markov CAC étiquettent un segment de trames en ayant une vue sur le contexte de ce segment. Ainsi, ce modèle semble adéquat à la segmentation et la reconnaissance des séquences de gestes. Cependant, Kasteren et al. ont montré dans [136] que la différence de performances de reconnaissance des activités humaines n'est pas importante entre le modèle CAC et le modèle semi-Markov CAC. De plus, Kasteren a constaté que l'étape d'apprentissage des modèles semi-Markov CAC est très longue à cause de l'étape d'estimation du facteur de normalisation du noyau CAC et de la prise en compte de toutes les possibilités de durée des segments des données. En ce qui concerne les CACDL, ces modèles étiquettent localement les trames en ayant une vue sur le contexte de la donnée de chaque trame grâce au principe du noyau CAC, et en ayant en même temps une vue globale sur la structure apprise de la donnée grâce aux états cachés. Ce modèle semble parfaitement adéquat à la reconnaissance des données séquentielles telles que les gestes mais reste couteux au niveau de l'application étant donné la complexité du modèle et le nombre élevé de paramètres à déterminer.

Nous détaillons dans la sous-section suivante, les avantages et les inconvénients de CAC.

4.3.6 Avantages et inconvénients des CAC

Les CAC sont des modèles discriminants. Ils ont la capacité de classer efficacement les données quelle que soit leur distribution. En effet, les modèles discriminant apprennent à discriminer les classes contrairement aux modèles génératifs, comme les MMC, qui apprennent à modéliser les classes. De plus, la décision locale des CAC dépend d'un contexte pris en compte sur toutes les observations contraire-

ment aux MMC dont la dépendance est limitée à l'observation locale. Cette propriété explique l'avantage des CAC au niveau de la classification par rapport aux MMC.

Néanmoins, les CAC classiques ne sont pas conçus pour modéliser une grammaire globale ou en général une structure globale des données. Ils sont des modèles de classification locale selon un processus séquentiel. Ainsi, la connaissance de haut niveau doit être introduite en post-traitement sous forme d'une étape supplémentaire de filtrage et de segmentation de l'information pour obtenir la cohérence structurelle des données qui est souvent recherchée. En contrepartie, les MMC possèdent cette capacité de segmentation et de structuration globale des données reconnues.

Finalement, si nous comparons les avantages et les inconvénients des CAC et des MMC, nous constatons une certaine complémentarité de classification et de segmentation entre les deux modèles. Ainsi, nous proposons de combiner ces deux modèles markoviens classiques afin d'exploiter simultanément les avantages des deux modèles et compenser leurs inconvénients. Nous proposons dans le prochain chapitre un modèle hybride CAC/MMC qui réalise cette combinaison. Nous donnons dans la prochaine section un aperçu sur les modèles hybrides combinant les MMC à d'autres méthodes de classification appliqués dans différents domaines.

4.4 Les modèles hybrides combinant les MMC à des méthodes de classification

La combinaison des MMC avec d'autres méthodes de classification n'est pas nouvelle. Les premières combinaisons de ce type ont été proposées dans les années 1990 avec l'intégration des réseaux de neurones au MMC [134]. Cette combinaison est la plus fréquente dans la littérature dans différents domaines. Ce type de modèle hybride a été appliqué à la reconnaissance de la parole [92, 99, 131, 65, 115, 158], la reconnaissance de l'écriture [132, 13, 72, 87, 48, 93, 88] et la reconnaissance des gestes [33]. Les MMC ont été également combinés aux SVM pour la reconnaissance de l'écriture [44] et aux méthodes de programmation dynamique pour la reconnaissance des gestes [111]. Nous constatons que l'application de ces modèles hybrides à la reconnaissance des gestes est récente et peu développée.

Concernant la combinaison des CAC et des MMC, à notre connaissance, seuls les travaux très récents de Soullard et al., en se basant sur les travaux de Gunawardana et al., ont abordé cette question [126, 53]. Dans ces travaux les auteurs contraignent l'apprentissage d'un CAC caché en l'initialisant avec les paramètres d'un MMC préalablement entraîné. Cette méthode assure la convergence de l'apprentissage du CAC caché et démontre la difficulté à maîtriser l'apprentissage de ce type de modèles. L'idée de notre approche est différente et s'inspire des approches neuro-Markoviennes de la littérature. Le principe de ces approches est de remplacer le modèle d'attache aux données du MMC constitué d'un mélange de

Gaussiennes, par un modèle discriminant qui classifie localement les observations. Ce modèle est traditionnellement constitué d'un réseau de neurones qui fournit la probabilité *a posteriori* de chaque classe associée à chaque observation locale dans la séquence. C'est cet étage discriminant que nous nous proposons de réaliser ici à l'aide d'un CAC. Le CAC aura pour tâche de discriminer les observations locales et fournira au MMC les probabilités *a posteriori* qui seront analysées pour un étiquetage de la séquence d'observations en tenant compte à la fois de la discrimination locale et du modèle global de la solution recherchée pris en compte par le modèle de transition du MMC. Selon le principe de notre modèle hybride, les MMC et les CAC sont appris séparément. Les détails de ce nouveau modèle hybride que nous proposons seront présentés dans le prochain chapitre.

Conclusion

Nous avons présenté dans ce chapitre le principe théorique de deux modèles markoviens de reconnaissance dynamiques : les MMC et les CAC. Nous avons déduit une complémentarité entre les deux modèles. Ainsi, nous proposons un modèle hybride combinant les avantages des MMC et des CAC tout en gardant un niveau de complexité raisonnable. Nous détaillerons dans le prochain chapitre le principe de notre modèle hybride, CAC/MMC, et nous montrerons sa robustesse sur les expérimentations effectuées.

Chapitre 5

Un modèle hybride, les CAC/MMC, pour la reconnaissance des gestes

Table des matières

Introduction

La reconnaissance des gestes combine deux tâches : la segmentation et la classification. Nous avons montré dans le chapitre précédent qu'il existe deux modèles Markoviens, les MMC et les CAC, qui ont l'aptitude à réaliser ces deux tâches. L'avantage principal des MMC est l'instance d'un modèle de transition leur permettant de structurer et segmenter les données. D'autre part, l'avantage principal des CAC est leur caractère discriminant qui prend en compte le contexte de chaque observation afin de mieux classifier les données localement. Ainsi, pour garantir l'efficacité de la tâche complète de reconnaissance, nous proposons une combinaison de ces deux modèles similairement aux modèles neuro-Markoviens. Pour évaluer les performances de notre système hybride et montrer sa robustesse au cas pratique de manque de données, nous avons utilisé les bases de données fournies dans le cadre d'une compétition de reconnaissance de gestes avec un seul exemple d'apprentissage, appelé également « *one-shot learning* ». Cette compétition est réalisée sur deux années 2011-2012 sous le nom de « *Gesture Challenge 1-2* » et éditée par l'organisation ChaLearn. Nous n'avons pas participé à cette compétition mais nous avons pu comparer notre système avec les travaux des participants grâce à la plateforme de gestion des participants proposée pour cette compétition [1]. Contrairement à notre travail de suivi des gestes, les bases de données, fournies dans le cadre de cette compétition, nous ont permis de varier les domaines des gestes (langue des signes, commandes robot, jeux vidéo...) et d'appliquer notre

1. https ://www.kaggle.com/c/GestureChallenge2

travail de reconnaissance sur des gestes quelconques. Nous désignons dans la suite la compétition « Gesture Challenge 1-2 » par la compétition ChaLearn.

Dans ce chapitre 5, nous présentons un nouveau système hybride de reconnaissance de gestes. Ce système combine les avantages des MMC et des CAC, et compense en même temps les inconvénients de ces deux systèmes classiques. Nous commençons par détailler le principe de notre modèle hybride CAC/MMC, et expliquerons son intérêt dans la section 5.1. Nous décrirons ensuite, dans la section 5.2, notre modèle de caractérisation des gestes en insistant sur sa particularité. Le modèle de caractérisation des gestes que nous avons conçu est plus large que le suivi d'une main. C'est un modèle de caractérisation globale introduisant d'autres techniques d'extraction d'informations. Puis, dans la section 5.3, nous expliquerons comment nous avons adapté notre système hybride à l'apprentissage avec un seul exemple. Ce cadre applicatif permet de concevoir un système de reconnaissance robuste au manque de données d'apprentissage. Nous exposerons enfin, dans la section 5.4, le protocole d'expérimentation, les résultats d'évaluation du système hybride, ainsi qu'un ensemble d'études effectuées dans le même cadre avec différents systèmes de reconnaissance. Nous terminons ce chapitre par la présentation des méthodes de vérification statistique adoptées dans la section 5.6.

5.1 Un modèle hybride de reconnaissance : les CAC/MMC

5.1.1 Avantages de la combinaison des CAC et des MMC

Nous rappelons que la reconnaissance de gestes nécessite deux tâches : la segmentation et la classification. Comme nous l'avons mentionné dans le chapitre précédent, les MMC sont bien adaptés à la tâche de segmentation grâce à leur modèle de transition. En revanche, les CAC s'adaptent mieux à la tâche de classification locale grâce à leur caractère discriminant.

En effet, le caractère discriminant des CAC leur permet de classer efficacement les observations même avec un manque de données d'apprentissage contrairement aux MMC qui sont des systèmes génératifs et leur modèle d'attache aux données se base sur l'estimation d'une densité de probabilité. En général, ces modèles génératifs nécessitent un grand nombre d'exemples d'apprentissage pour estimer correctement les densités. De plus, les CAC permettent de tenir compte du contexte de chaque observation à travers les fonctions de caractéristiques pour la classer. Dans la pratique, le contexte est défini par une fenêtre présentant un hyperparamètre : la largeur de la fenêtre d'observation. Par la suite, le décodage des données prend en compte leurs dépendances locales renforçant la classification des trames.

Ainsi, la combinaison du caractère discriminant des CAC à travers les probabilités *a posteriori* et la modélisation structurelle globale des données par les MMC à travers les probabilités de transitions présente le point fort de notre modèle hybride.

Dans le cas de la reconnaissances de séquences de gestes, nous constatons que la conception d'un modèle global parallèle de MMC est fréquente dans la littérature [141, 142, 101, 73, 62, 146]. Dans notre cas, nous reprenons la même idée conceptuelle en mettant en compétition toutes les classes de gestes, autrement dit en liant parallèlement les modèles de transition MMC des différents gestes du vocabulaire. Nous détaillons ce modèle global dans la sous-section 5.1.2. Nous montrerons dans les sous-sections 5.5.3.2 et 5.5.3.1 que l'intégration des CAC nous permet de prendre en compte l'élasticité de la durée des gestes sans avoir recours à un modèle global ergodique de transitions complexe avec un grand nombre d'états, des connexions complètes et des sauts entre états comme dans le cas du modèle de la figure 4.4 (chapitre précédent) proposée dans [73, 62, 146].

5.1.2 Présentation du modèle CAC/MMC

Dans cette sous-section nous présentons le modèle hybride proposé pour la reconnaissance de gestes. Ce modèle est basé sur la combinaison des probabilités *a posteriori* locales calculées par le CAC et des probabilités de transitions globales entre les classes calculées par le MMC. Nous rappelons que ce système hybride permet de combiner la capacité des MMC de modéliser la continuité d'un processus séquentiel et la capacité des CRF de discriminer les données observées localement avec robustesse. La figure 5.1 clarifie graphiquement ce nouveau modèle hybride combinant les MMC et les CAC. Les MMC garantissent la prise en compte du contexte liant les classes et les CAC garantissent la prise en compte du contexte liant les observations.

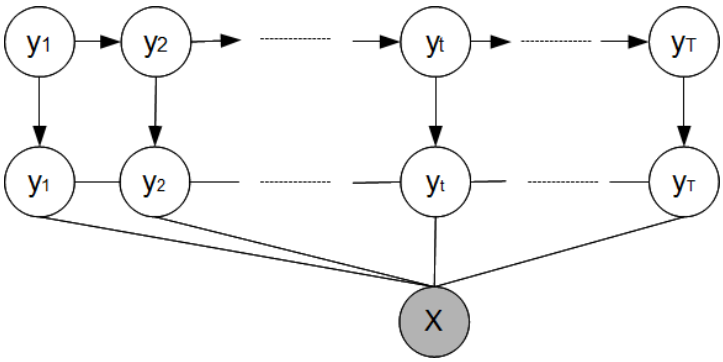


FIGURE 5.1 – Le modèle graphique de notre système hybride CAC/MMC

En effet, la décision locale des CAC est injectée dans le modèle contextuel des MMC à travers l'algorithme de décodage Viterbi. Cette décision vient remplacer la probabilité d'observation calculée par les mélanges des gaussiennes $p(x_t|y_t)$. Ainsi, la probabilité $p(y_{1:T}, x_{1:T})$ (équation 5.1), modélisée par le MMC, dépend de la probabilité *a posteriori* locale calculée par le CAC.

$$p(y_{1:T}, x_{1:T}) = p(x_1|y_1)p(y_1)\prod_{t=2}^{T} p(x_t|y_t)p(y_t|y_{t-1}) \tag{5.1}$$

Or $p(x_t|y_t)$ est une vraisemblance et les CAC fournissent une probabilité *a posteriori* $p(y_t|x_t)$. Ainsi, il faut écrire $p(x_t|y_t)$ en fonction de $p(y_t|x_t)$. En effet, la formule de Bayes donne :

$$p(x_t|y_t) = \frac{p(y_t|x_t)p(x_t)}{p(y_t)} \tag{5.2}$$

Toutes les classes ou sous-classes des gestes y sont équiprobables donc $p(y_t)$ est une constante $\forall t \in \mathbb{N}$. Puisque le but est de trouver la séquence $y_{1:T}$ qui maximise $p(y_{1:T}, x_{1:T})$ et puisque la probabilité d'observation des trames $p(x_t)$ est indépendante du temps, $p(x_t)$ n'intervient pas dans le processus de maximisation de $p(x_t|y_t)$. Ainsi pour maximiser $p(x_t|y_t)$ il suffit de maximiser $p(y_t|x_t)$.

Étant donné que les CRF permettent de prendre en compte la totalité des observations pour calculer à chaque instant la probabilité de chaque classe, nous pouvons écrire $p(y_t|x_t) = p(y_t|x_{1:T})$. Nous rappelons que $y_{1:T}$ est noté Y et $x_{1:T}$ est notons X.

Cette quantité est modélisé par un CAC et calculée avec l'algorithme forward-backward [9]. Traditionnellement, on note la probabilité "forward" α_t et la probabilité "backward" β_t. Ces deux variables sont calculées à travers une relation de récurrence :

$$\alpha_t(i) = p(x_1 x_2 .. x_t, y_t = s_i) = \sum_{j=1}^{N_s} \alpha_{t-1}(j)\psi_t(s_i, s_j, o_l) \tag{5.3}$$

$$\beta_t(i) = p(x_{t+1} x_{t+2} .. x_T, y_t = s_i) = \sum_{j=1}^{N_s} \beta_{t+1}(j)\psi_{t+1}(s_i, s_j, o_l) \tag{5.4}$$

où

$$\psi_t(s_i, s_j, o_l) = \exp(\sum_{k=1}^{K} \lambda_k f_k(y_t = s_i, y_{t-1} = s_j, x_t = o_l)) \tag{5.5}$$

et s_i, s_j sont deux valeurs d'états cachés appartenant à $\mathcal{S}$ et o_l est une valeur d'observation appartenant à $\mathcal{O}$. Enfin, d'après le processus "forward-backward" nous avons :

$$p(X) = \sum_{j=1}^{N_s} \alpha_T(j) = \sum_{j=1}^{N_s} \beta_1(j) = \sum_{j=1}^{N_s} \alpha_t(j)\beta_t(j) \tag{5.6}$$

$$p(y_t = s_i|X) = \frac{p(y_t = s_i, X)}{p(X)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N_s} \alpha_t(j)\beta_t(j)} = \gamma_t(i) \tag{5.7}$$

L'apprentissage des composantes MMC et CAC est réalisé séparément. La figure 5.2 représente les différentes étapes d'apprentissage et de décodage avec le système hybride CAC/MMC. L'apprentissage des MMC est réalisé en premier lieu. Les MMC appris, correspondants aux différentes classes de gestes, sont utilisées pour étiqueter la base d'apprentissage nécessaire pour l'apprentissage du CAC. Les CAC sont appris en deuxième lieu. En phase de reconnaissance une première étape de décodage Forward-Backward est réalisée avec le modèle CAC. Elle permet d'inférer les probabilités *a posteriori* locales. Une seconde étape de décodage Viterbi, réalisée avec le MMC en utilisant les probabilité *a posteriori* calculées avec les CAC, permet de décoder la meilleur séquence d'étiquettes correspondant à la séquence vidéo donnée. Nous détaillerons davantage l'étape d'apprentissage dans la sous-section 5.1.4 et l'étape de décodage dans la sous-section 5.1.5. Nous présenterons dans la sous-section suivante les modèles des composantes élémentaires MMC et CAC.

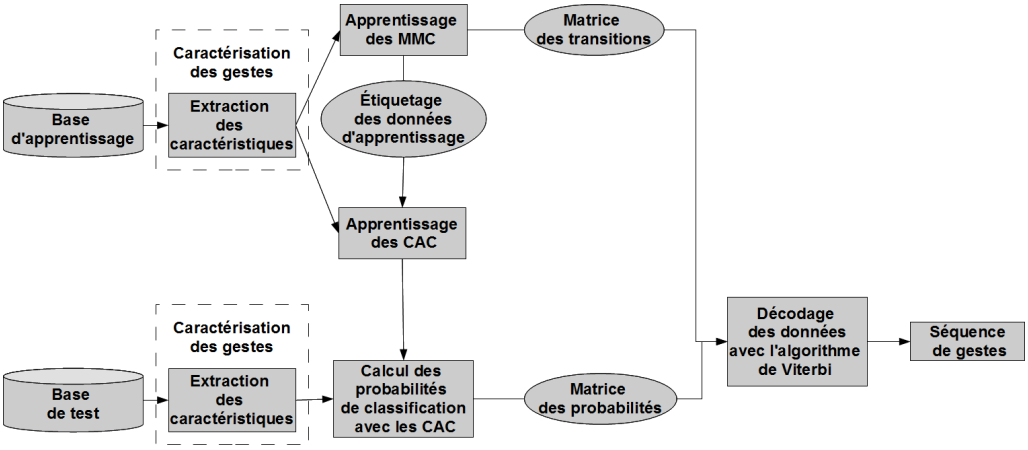


FIGURE 5.2 – Les différentes étapes d'apprentissage et de décodage des gestes avec les CAC/MMC

5.1.3 Architecture générale des CAC/MMC

Le modèle de la composante MMC comporte deux niveaux de granularité : le niveau geste et le niveau séquence de gestes. L'étape d'apprentissage génère pour chaque geste un modèle MMC gauche droite. Le nombre d'états des modèles de gestes varie selon la taille du geste. Afin d'adapter le système à la variabilité de la durée des gestes, nous avons choisi un nombre d'états variable pour les modèles MMC des gestes. Nous avons vérifié empiriquement que les performances de reconnaissance du système avec un nombre d'états variable selon la taille du geste dépassent les performances de reconnaissance avec un nombre d'états fixe pour tous les gestes. Le nombre d'états de chaque geste i est déterminé automatiquement selon sa taille $\mathtt{f_g}(i)$. Le nombre théorique de trames par état, que nous notons $\mathtt{f_e}$, est un hyper-paramètre du système. Nous notons le nombre d'états

pour un geste i ; $\mathtt{N_e}(i) = \frac{\mathtt{f_g}(i)}{\mathtt{f_e}}$. Le nombre de gaussiennes est choisi selon le cadre applicatif expliqué dans la sous-section 5.3.1.

La composante CAC a une structure linéaire classique telle que modélisée dans la figure 5.1. L'étape d'apprentissage génère un seul système CAC pour toutes les classes de gestes à discriminer. Les CAC ont une fenêtre d'observation permettant de prendre en compte l'observation courante et son contexte. Afin d'adapter davantage le système à la variabilité de la durée des gestes, nous choisissons une taille variable $\mathtt{f_w}$ de cette fenêtre d'observation $\mathtt{w_o}$ dépendant d'une taille représentative du vocabulaire de gestes $\mathbb{G}$ (équation 5.8). Autrement dit, la taille de la fenêtre d'observation $\mathtt{f_w}(\mathtt{w_o}(\mathbb{G}))$ est choisie égale à $\frac{3}{4}$ de la taille du geste le plus court de la base d'apprentissage. Ce choix est validé empiriquement.

$$\mathtt{f_w}(\mathtt{w_o}(\mathbb{G})) = \min(\frac{3}{4} \min_{\mathtt{g} \in \mathbb{G}} \mathtt{f_g}(\mathtt{g}), \text{seuil}) \tag{5.8}$$

5.1.4 Apprentissage

L'apprentissage des MMC et l'apprentissage des CAC se font séparément. L'apprentissage des MMC génère la matrice des transitions entre les états des gestes que nous appelons également les sous-classes des gestes ou les sous-unités. Pour chaque geste, un modèle de transitions est appris séparément, puis nous générons un modèle de transitions global pour tous les gestes. Nous présenterons ce modèle dans la sous-section 5.1.5.

Les CAC ne disposent pas d'un apprentissage embarqué comme les MMC. Leur apprentissage nécessite un étiquetage par trame de la base d'apprentissage, il nécessite ainsi une vérité terrain élémentaire. Par conséquent, après l'apprentissage des MMC, un alignement forcé sur la base d'apprentissage est effectué pour préparer la vérité terrain pour les CAC. L'étape d'apprentissage des CAC génère un seul modèle pour tout les gestes en considérant que chaque geste est un ensemble de sous-classes. Pour chaque geste, le nombre de sous-classes est égale au nombre d'états de son modèle MMC.

D'autre part, l'algorithme d'apprentissage des CAC a un hyper-paramètre de régularisation permettant à un système d'éviter le sur-apprentissage ou le sous-apprentissage. Une valeur élevée de ce paramètre peut induire le système en sur-apprentissage. Notre système semble être équilibré avec une valeur de 1.5 que nous avons déterminée empiriquement.

5.1.5 Décodage

L'étape de décodage d'une séquence de trames avec les CAC génère pour chaque trame, t, un vecteur de probabilités *a posteriori*, $p(y_t = s_i|X)$ pour toute sous-classe s_i correspondant à un état des MMC. Les CAC classiques déterminent à partir de ce vecteur la décision locale. Dans le cas de notre modèle hybride, nous récu-

pérons toutes les probabilités *a posteriori* pour toutes les trames et nous avons ainsi une matrice P de taille $\mathtt{f_q(s)} \times \mathtt{N_e}(\mathbb{G})$, où $\mathtt{f_q(s)}$ est le nombre de trames de la séquence de gestes `s`.

Concernant l'étape de décodage globale du système CAC/MMC, le système doit être capable de décoder des données contenant un séquencement de gestes en ordre aléatoire. Ainsi, nous avons conçu un modèle global reliant tous les modèles MMC élémentaires des gestes avec des transitions inter-gestes équiprobables. La figure 5.3 montre une représentation simplifiée du modèle global. Il s'agit d'un MMC parallèle où chaque ligne contient un modèle MMC de geste élémentaire généré à l'étape d'apprentissage. Le système peut transiter de l'état de début `d` d'une manière équiprobable vers le premier état de l'un des gestes. Du dernier état de chaque geste $e_{n_i}^{g_i}$, où n_i représente le nombre d'états du geste i, le système peut transiter vers le premier état du même geste ou vers le premier état de l'un des autres gestes ou vers l'état final `f` d'une manière équiprobable. Le séquencement des gestes est aléatoire, il ne répond pas à une grammaire de gestes bien précise. Ce modèle globale peut être traduit par une matrice de transitions T de taille $\mathtt{N_e}(\mathbb{G}) \times \mathtt{N_e}(\mathbb{G})$, où $\mathbb{G}$ est le vocabulaire des gestes et $\mathtt{N_e}(\mathbb{G})$ est le nombre d'états correspondant à tous les gestes du vocabulaire.

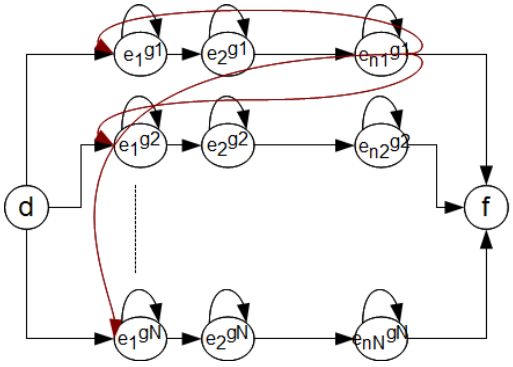


FIGURE 5.3 – Le modèle de reconnaissance des séquences de gestes conçu avec les MMC. $e_j^{g_i}$ représente l'état j du geste i.

Enfin, l'algorithme de Viterbi effectue le décodage final de la séquence en exploitant les deux matrices T et P. Ainsi, toutes les probabilités *a posteriori* fournies par les CAC sont exploitées selon une cohérence structurelle globale imposée par les MMC. En effet, les CAC peuvent donner une probabilité d'une sous-classe qui ne soit pas maximale localement mais qui soit suffisamment importante et suffisamment cohérente avec le modèle de transitions pour qu'elle appartienne au chemin final généré par l'algorithme de Viterbi correspondant à la séquence d'étiquettes la plus probable.

Pour réaliser tout ce processus de reconnaissance de gestes, une étape préliminaire de caractérisation des gestes est indispensable comme le montre la figure 5.2.

Dans la section suivante, nous allons présenter notre modèle de caractérisation des gestes.

5.2 Caractérisation globale des gestes

Les caractéristiques des gestes doivent minimiser la variabilité intra-classes et maximiser la variabilité inter-classes. Ainsi, le système de caractéristiques doit être le plus complet possible et ses composantes doivent être les plus complémentaires possible. Or, la méthode de caractérisation des gestes par le suivi, proposée dans la première partie de ce manuscrit, est une méthode de caractérisation locale basée sur le suivi d'un point d'intérêt qui est la main dominante dans notre cas. Ainsi, pour compléter notre modèle de caractérisation, il s'avère nécessaire d'ajouter des caractéristiques de description globale des gestes telles que la description de l'évolution spatiotemporelle de la totalité du mouvement corporel et la description de l'évolution spatiale des contours. Ces nouvelles caractéristiques nous permettent de décrire des situations de communication gestuelle plus générales que celles analysées dans la première partie de ce document et qui était uniquement basées sur le mouvement de la main dominante.

Dans cette section nous proposons un ensemble de descripteurs de mouvement que nous avons conçus à partir des vitesses calculées avec les flots optiques. Nous appelons l'ensemble de ces descripteurs « la Signature du Geste » (`SG`). Nous proposons également d'ajouter des descripteurs de forme extraits avec les Histogramme de Gradients Orientés (`HOG`).

5.2.1 Caractérisation avec les Flots Optiques : Signature du Geste

Les flots optiques sont des extracteurs de vitesse au niveau pixel, connus pour leur robustesse aux variations de luminosité [15]. Ils sont invariants aux couleurs et aux déformations des objets. Le principe des flots optiques est présenté dans la sous-section 2.5.3 du chapitre 2.

Les flots optiques et les filtres particulaires permettent tous deux de caractériser le mouvement d'un geste. Les différences entre ces deux techniques sont les suivantes ; premièrement, les flots optiques sont capables de décrire simultanément tous les mouvements existant dans la scène contrairement aux filtres particulaires qui suivent des objets précis de la scène, deuxièmement, les filtres particulaires nécessitent une étape supplémentaire de caractérisation de l'objet suivi. Ainsi, les flots optiques semblent adéquats pour extraire simultanément un maximum d'information sur les mouvements corporels constituant un geste tout en étant robustes aux variabilités de couleurs, de forme et de luminosité.

Nous proposons dans la suite un vecteur de caractéristiques dont les composantes sont des combinaisons des valeurs des vitesses calculées par les flots optiques. Nous appelons ce vecteur de caractéristiques `SG`. Puisque les mains sont

les membres principaux établissant le geste, ce vecteur décrit essentiellement leur mouvement.

Comme le mouvement des mains est localisé généralement sur la partie gauche et la partie droite de l'image, il est avantageux de diviser l'image en deux parties verticalement comme montré dans la figure 5.4. Ainsi, la description du mouvement est localisée davantage, et l'on peut caractériser les mouvements dans ces deux régions distinctes.

Dans certaines expérimentations effectuées et présentées plus loin dans ce document, le vecteur SG n'est pas calculé sur les deux parties de l'image mais sur la totalité. Dans cette situation SG contient 18 composantes. Les 18 composantes sont issues de 9 descripteurs appliqués sur les composantes horizontales, $\mathtt{V}_\mathtt{X}^+$ et $\mathtt{V}_\mathtt{X}^-$, et sur les composantes verticales, $\mathtt{V}_\mathtt{Y}^+$ et $\mathtt{V}_\mathtt{Y}^-$, des vitesses des pixels de l'image calculées par les flots optiques (figure 5.4).

L'extraction des flots optiques se fait à partir de deux images successives en calculant des intensités comme expliqué dans la sous-section 2.5.3. Pour chaque pixel de ces deux images un vecteur vitesse est calculé et retourné sous forme de deux composantes $\mathtt{V}_\mathtt{X}$ et $\mathtt{V}_\mathtt{Y}$. Chaque composante peut avoir un signe négatif ou positif ce qui correspond à un sens positif ou négatif du pixel. Nous les notons $\mathtt{V}_\mathtt{X}^+(p)$, $\mathtt{V}_\mathtt{X}^-(p)$, $\mathtt{V}_\mathtt{Y}^+(p)$ et $\mathtt{V}_\mathtt{Y}^-(p)$, où p est un pixel de l'image. Évidement, pour chaque pixel p, deux de ces valeurs sont nulles, un pixel ne peut avoir qu'un seul sens selon l'axe des x et qu'un seul sens selon l'axe des y. Notre `signature` du geste est un ensemble de fonctions combinant ces composantes de vitesses de manière à extraire différentes informations énergétiques et spatiales. Ces informations sont représentées par les 9 descripteurs mentionnés plus haut. Ces 9 descripteurs sont composés de 4 descripteurs de *localisation* du mouvement, 2 descripteurs de *vélocité* du mouvement et 3 descripteurs *d'orientation* du mouvement. Ces descripteurs sont calculés selon l'axe horizontal et l'axe vertical ce qui multiplie leur nombre par 2 (ce qui donne 18 caractéristiques), et par 4 (ce qui donne 36 caractéristiques) dans le cas où l'on divise l'image en deux régions.

Le tableau 5.1 présente les 8 caractéristiques de localisation (8 = 4 sur $\mathtt{V}_\mathtt{X}$ + 4 sur $\mathtt{V}_\mathtt{Y}$) analogues à des coordonnées de centres d'inertie. Elles représentent les positions verticales et horizontales des centres de vitesse liés au mouvement.

Le tableau 5.2 présente les 4 caractéristiques de vélocité ou force du mouvement (4 = 2 sur $\mathtt{V}_\mathtt{X}$ + 2 sur $\mathtt{V}_\mathtt{Y}$). Le premier descripteur présente une information énergétique du mouvement. Il est inversement proportionnel à la moyenne quadratique des vitesses des pixels en mouvement. Pour des raisons de normalisation, nous utilisons l'inverse de cette moyenne. Le deuxième descripteur présente une information sur l'amplitude du mouvement. C'est la médiane des vitesses des pixels en mouvement. La médiane permet d'intégrer une information sur la quantité du mouvement, où la masse est remplacée dans notre cas par le nombre des pixels qui bougent. Par exemple, si le nombre de pixels qui ont une vitesse éle-

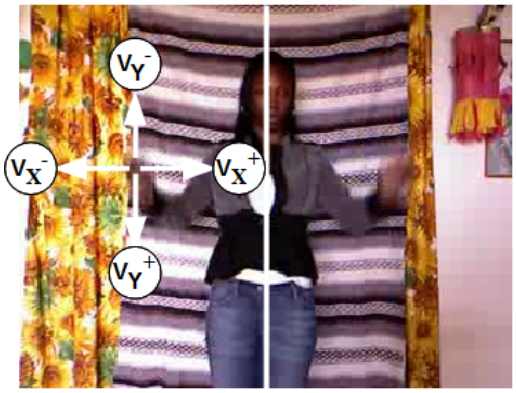


FIGURE 5.4 – Les directions des composantes des flots optiques (image d'une vidéo des bases de données `ChaLearn`)

Descripteur	horizontalement	verticalement
abscisse moyen des pixels se déplaçant dans le sens positif	$\frac{1}{I_w} \times \frac{\sum_{p=1}^{N_{px}^+} \lvert V_X^+(p)\rvert x_p}{\sum_{p=1}^{N_{px}^+} \lvert V_X^+(p)\rvert}$	$\frac{1}{I_w} \times \frac{\sum_{p=1}^{N_{px}^+} \lvert V_Y^+(p)\rvert x_p}{\sum_{p=1}^{N_{px}^+} \lvert V_Y^+(p)\rvert}$
ordonnée moyenne des pixels se déplaçant dans le sens positif	$\frac{1}{I_h} \times \frac{\sum_{p=1}^{N_{px}^+} \lvert V_X^+(p)\rvert y_p}{\sum_{p=1}^{N_{px}^+} \lvert V_X^+(p)\rvert}$	$\frac{1}{I_h} \times \frac{\sum_{p=1}^{N_{px}^+} \lvert V_Y^+(p)\rvert y_p}{\sum_{p=1}^{N_{px}^+} \lvert V_Y^+(p)\rvert}$
abscisse moyen des pixels se déplaçant dans le sens négatif	$\frac{1}{I_w} \times \frac{\sum_{p=1}^{N_{px}^-} \lvert V_X^-(p)\rvert x_p}{\sum_{p=1}^{N_{px}^-} \lvert V_X^-(p)\rvert}$	$\frac{1}{I_w} \times \frac{\sum_{p=1}^{N_{px}^-} \lvert V_Y^-(p)\rvert x_p}{\sum_{p=1}^{N_{px}^-} \lvert V_Y^-(p)\rvert}$
ordonnée moyenne des pixels se déplaçant dans le sens négatif	$\frac{1}{I_h} \times \frac{\sum_{p=1}^{N_{px}^-} \lvert V_X^-(p)\rvert y_p}{\sum_{p=1}^{N_{px}^-} \lvert V_X^-(p)\rvert}$	$\frac{1}{I_h} \times \frac{\sum_{p=1}^{N_{px}^-} \lvert V_Y^-(p)\rvert y_p}{\sum_{p=1}^{N_{px}^-} \lvert V_Y^-(p)\rvert}$

TABLE 5.1 – Les caractéristiques de **localisation** du mouvement

vée est élevé la médiane va avoir une valeur élevée, cette valeur va être réduite dans le cas contraire, contrairement à la moyenne qui dépend fortement de la valeur maximale de la vitesse dans l'image. La médiane permet également d'atténuer l'effet du bruit. Les composantes $\mathtt{V}^*_{\mathtt{X}}$ et $\mathtt{V}^*_{\mathtt{Y}}$ sont les médianes d'un vecteur de vitesses seuillées et calculées avec les flots optiques. En effet, nous utilisons un terme S_V ($S_{V_X} = \frac{\sum_{p=1}^{\mathtt{N}^s_{\mathtt{px}}} |\mathtt{V}_{\mathtt{X}}(p)|}{\mathtt{N}^s_{\mathtt{px}}}$, $S_{V_Y} = \frac{\sum_{p=1}^{\mathtt{N}^s_{\mathtt{px}}} |\mathtt{V}_{\mathtt{Y}}(p)|}{\mathtt{N}^s_{\mathtt{px}}}$) pour seuiller toutes les valeurs de vitesse prises en compte dans le calcul de toutes les caractéristiques de mouvement. Ce seuillage permet de réduire le bruit et les mouvements négligeables qui n'interviennent pas dans l'élaboration du geste tel que le mouvement de la tête et le mouvement du buste. Nous utilisons également ce seuil pour normaliser la médiane.

Descripteur	horizontalement	verticalement
L'inverse d'une **vitesse globale**	$\frac{1}{\sqrt{\frac{\sum_{p=1}^{\mathtt{N}_{\mathtt{px}}} (\mathtt{V}_{\mathtt{X}}(p))^2}{\mathtt{N}_{\mathtt{px}}}}}$	$\frac{1}{\sqrt{\frac{\sum_{p=1}^{\mathtt{N}_{\mathtt{px}}} (\mathtt{V}_{\mathtt{Y}}(p))^2}{\mathtt{N}_{\mathtt{px}}}}}$
médiane des vitesses maximales	$\frac{1}{S_{V_X}} \times \lvert\mathtt{V}^*_{\mathtt{X}}\rvert$	$\frac{1}{S_{V_Y}} \times \lvert\mathtt{V}^*_{\mathtt{Y}}\rvert$

TABLE 5.2 – Les caractéristiques de **vélocité** du mouvement

Le tableau 5.3 présente les 6 caractéristiques d'orientation du mouvement (6 = 3 sur $\mathtt{V}_{\mathtt{X}}$ + 3 sur $\mathtt{V}_{\mathtt{Y}}$). Ces caractéristiques sont des statistiques sur les pixels se déplaçant dans un sens commun, positif ou négatif, normalisé par leur effectif. Les deux premiers descripteurs caractérisent la quantité de pixels se déplaçant dans le même sens. Le troisième descripteur décrit l'orientation dominante du mouvement. Ce descripteur, associé au deux premiers, caractérise la relation ou la symétrie entre les deux groupes principaux de mouvement dont les orientations sont opposées. La figure 5.5 montre l'intérêt de ces descripteurs et illustre cette information de symétrie. Cette figure présente deux courbes de variation du nombre des pixels dans le sens positif et dans le sens négatif sur l'axe horizontal durant un extrait d'une vidéo contenant un discours en langue de signe de la base de données `SignStream` [97]. Nous constatons que lorsque les deux courbes se superposent avec la présence d'un pic, un mouvement opposé des deux mains se produit dans l'extrait des trames correspondant. Lorsque l'écart entre les deux courbes est élevé, un mouvement parallèle des deux mains dans le sens dominant se produit. La valeur de l'écart et le sens dominant sont déduits à partir de la valeur absolue et le signe du troisième descripteur. Lorsque les deux courbes stagnent, les deux mains sont fixes dans les trames associées (la trame 70 dans la figure présente une main fixe dont les doigts sont en mouvement léger épelant un nom selon les règles primaires de la langue des signes). Ainsi, en analysant la variation de ces trois descripteurs, nous pouvons déduire le type de mouvement associé. D'où l'importance et la complémentarité de ces trois descripteur d'orientation.

Descripteur	horizontalement	verticalement
proportion des pixels se déplaçant dans le sens positif	$\frac{N_{px}^{v_x^+}}{N_{px}}$	$\frac{N_{px}^{v_y^+}}{N_{px}}$
proportion des pixels se déplaçant dans le sens négatif	$\frac{N_{px}^{v_x^-}}{N_{px}}$	$\frac{N_{px}^{v_y^-}}{N_{px}}$
L'**orientation** dominante	$\frac{N_{px}^{v_x^+} - N_{px}^{v_x^-}}{N_{px}}$	$\frac{N_{px}^{v_y^+} - N_{px}^{v_y^-}}{N_{px}}$

TABLE 5.3 – Les caractéristiques de l'**orientation** du mouvement

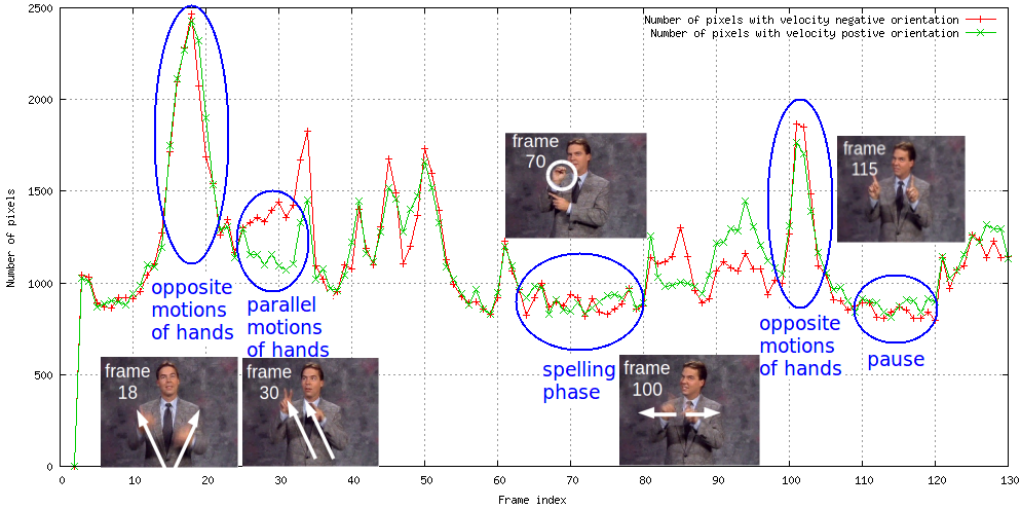


FIGURE 5.5 – Variation des nombres des pixels ayant une orientation commune (images de la langue des signes d'une vidéo de la base de données `SignStream`)

Enfin, bien que ces caractéristiques semblent simples, elles sont complémentaires et spécifient minutieusement l'évolution des gestes étant donné qu'elles décrivent la **localisation**, la **vélocité** et l'**orientation** du mouvement qui est une composante principale d'un geste. Nous allons confirmer cette affirmation par des résultats de reconnaissance dans la sous-section 5.5.1.

Pour avoir une caractérisation complète du geste, nous ajoutons des caractéristiques de contours globales extraites avec un descripteur de forme classique ; les Histogrammes de Gradients Orientés (`HOG`). Nous présentons cette caractérisation de forme dans la sous-section suivante.

5.2.2 Caractérisation avec les HOG

Comme notre but est de caractériser les gestes, nous exploitons les HOG pour décrire la pose du signeur du geste à chaque instant.

Les HOG sont des histogrammes d'orientation des gradients calculés sur une image afin de décrire les contours contenus dans cette image. Ils sont exploités principalement pour la détection des personnes [35]. Pour appliquer ce descripteur, nous avons repris l'implémentation de Dalal et al. [35].

La figure 5.6 présente les 9 directions adoptées pour quantifier les angles d'inclinaisons des gradients calculés sur l'image. Selon les travaux de Dalal et al [35], ces 9 orientations donnent de bons résultats pour la détection des personnes. Dans les HOG, ces 9 directions sont pondérées par les normes des gradients correspondants dans les cellules de calcul sur l'image.

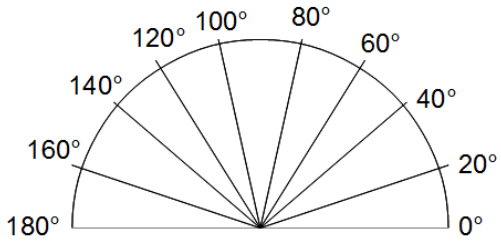


FIGURE 5.6 – Les niveaux de quantification des orientations adoptés pour les HOG

L'algorithme de calcul des HOG proposé par Dalal et al. [35] applique une auto-superposition de la fenêtre glissante, ce qui génère des histogrammes redondants et un vecteur de caractéristiques HOG de très grande taille (égale à 3780). Pour alléger ce vecteur, nous avons eu recours à l'application de l'opérateur moyenne sur deux niveaux. Le premier niveau de simplification est basé sur l'algorithme de visualisation des HOG proposé par Jürgen Brauer [2]. L'idée de cet algorithme est de moyenner les histogrammes redondants sur des cellules de l'image en gardant les 9 orientations des gradients dans chaque cellule. La figure 5.7 présente un exemple de visualisation des HOG fourni par Brauer, où l'amplitude du gradient pour chaque orientation est traduite par la longueur du trait correspondant dans chaque cellule de l'image.

Le deuxième niveau de simplification du descripteur HOG, que nous avons appliqué dans notre cas, est de moyenner les amplitudes du gradient sur des blocs plus large sur l'image que nous appelons des méta-blocs. Nous avons partitionné l'image en 4 méta-blocs puis en 16 méta-blocs [3]. Pour chaque méta-bloc et pour chaque orientation, nous avons appliqué la moyenne sur les amplitudes correspondantes dans toutes les cellules du méta-bloc. Enfin, nous obtenons 9 valeurs

2. http ://www.juergenwiki.de/work/wiki/doku.php ?id=public%3ahog_descriptor_computation_and_visualization
3. nous avons effectué des tests de reconnaissance avec chaque cas séparément

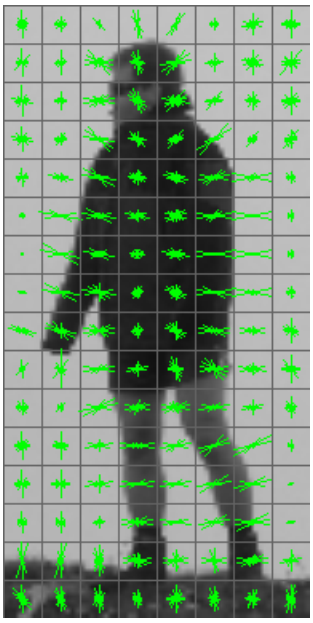


FIGURE 5.7 – Exemple de visualisation du descripteur HOG

d'amplitude multipliées par le nombre de méta-blocs, ce qui correspond à un descripteur de taille 36 (36 = 9 orientations × 4 méta-blocs) dans le premier cas et un descripteur de taille 144 (144 = 9 orientations × 16 méta-blocs) dans le deuxième cas. Enfin, nous appliquons les HOG sur la différence de deux images successives afin de retirer le font de l'image.

Nous résumons dans la sous-section suivante les variantes du vecteur de caractéristiques adoptées pour les tests réalisés.

5.2.3 Les différentes variantes du vecteur de caractéristiques

Le tableau 5.4 présente les différentes variantes du vecteur de caractéristiques $\vec{c}$ que nous avons utilisées dans nos tests de reconnaissances de gestes. Nous indexons chaque variante par sa taille $l(\vec{c})$. $l(\vec{c}(\texttt{SG}))$ est le nombre de caractéristiques utilisées issues de la signature du geste. $l(\vec{c}(\texttt{HOG}))$ est le nombre de caractéristiques utilisées issues des HOG. $l(\vec{c}(P))$ est le nombre de caractéristiques associées à la position de la main dominante estimée par les filtres particulaires. Certaines variantes du vecteur $\vec{c}$ sont appliquées sur deux formats de données (format RGB et format en profondeur).

taille totale ($l(\vec{c})$)	**Descripteur**			
	Signature du geste SG		HOG	
	$l(\vec{c}(\texttt{SG}))$	description	$l(\vec{c}(\texttt{HOG}))$	description
54	18	sans division de l'image en 2 régions	36	4 méta-blocs
52	16	sans la médiane, sans division de l'image en 2 régions	36	4 méta-blocs
180	36	division de l'image en 2 régions	144	16 méta-blocs
360	72	division de l'image en 2 régions, 2 formats de données	288	16 méta-blocs, 2 formats de données
72	72	division de l'image en 2 régions, 2 formats de données	0	HOG non appliqués

$l(\vec{c})$	**Descripteur**	
	Position de la main dominante	
	$l(\vec{c}(P))$	description
4	4	abscisse, ordonnée, largeur, longueur

TABLE 5.4 – Les différentes variantes du vecteur caractéristiques adoptées dans nos expérimentations

5.3 Adaptation des CAC/MMC à l'apprentissage avec un seul exemple

Afin de concevoir un système de reconnaissance robuste au manque de données d'apprentissage, nous choisissons un cadre applicatif d'apprentissage avec un seul exemple. Ce cadre s'avère très pratique dans le cas réel puisqu'il analyse le comportement d'un système lorsqu'on le paramétrise avec le minimum d'effort d'annotation des données en demandant à l'utilisateur de n'annoter qu'un seul exemple pour le fournir au système, ce qui explique la motivation de la compétition ChaLearn (*Gesture Challenge : one shot-learning*). De plus, l'utilisation des bases de données de petite taille accélère l'apprentissage et facilite l'exploitation des systèmes de reconnaissance avec un apprentissage en temps réel.

Certains travaux ont été élaborés dans ce cadre d'apprentissage avec un seul exemple [73, 62, 146, 149, 153]. La plupart de ces systèmes se basent sur des modèles classiques de reconnaissance avec une adaptation au cas d'apprentissage

avec un seul exemple. Nous reprenons certains de ces travaux dans la section 5.5 dans un cadre de comparaison.

5.3.1 Adaptation de la composante des MMC

Pour modéliser l'espace des caractéristiques, les MMC classiques utilisent des mélanges de gaussiennes en se basant sur les exemples fournis. Les centres et les variances des gaussiennes dépendent de la distribution de ces exemples dans l'espace de caractéristiques. Quand le nombre d'exemples est très réduit, il est très difficile d'estimer précisément ces gaussiennes et en particulier leurs variances. Ainsi, dans un premier temps, nous avons réduit le nombre de gaussiennes à une seule gaussienne par classe de gestes. Nos premières expériences ont bien montré l'effet positif de cette réduction du nombre de gaussiennes au niveau des performances de la reconnaissance des gestes. Dans un second temps, nous avons régularisé les variances de chaque gaussienne à la valeur moyenne pour tous les gestes. Cette modification représente une étape de post-traitement succédant l'étape d'apprentissage. Ce post-traitement permet d'unifier les valeurs des variances puisque les exemples d'apprentissage ne sont pas assez nombreux pour déterminer correctement ces valeurs. Nos expériences ont montré également l'amélioration des performances avec cette adaptation du modèle MMC (section 5.5).

5.3.2 Adaptation de la composante des CAC

En ayant un seul exemple pour chaque classe et en ayant des caractéristiques continues, il est difficile d'optimiser les paramètres du système CAC. Ainsi, nous avons procédé à la quantification des valeurs des caractéristiques. Cette quantification permet de généraliser le système CAC appris et augmenter sa capacité de reconnaissance. En effet, en exploitant les CAC discrets, chaque plage de valeurs de chaque caractéristique aura un poids différent dans le vecteur des paramètres du système. La distinction des poids spécialise davantage les fonctions de caractéristiques et augmente le pouvoir discriminant des CAC. Dans le cas des CAC continus, à chaque caractéristique est attribué un seul poids. La même pondération est donnée aux valeurs faibles et aux valeurs élevées de la caractéristique. Or pour la même caractéristique, ses valeurs faibles peuvent être distinctives plus que les valeurs fortes. Ainsi, la pondération de chaque caractéristique avec le même poids ne semble pas significatif dans tous les cas. Les tests que nous avons réalisés ont montré que les performances des CAC discrets excèdent les performances des CAC continus. Nous donnons dans la sous-section 5.5.2 les résultats de ce test.

Pour réaliser cette quantification, nous avons utilisé un quantifieur scalaire uniforme symétrique (équation 5.9).

$$Q: \begin{array}{ccc} [-\mathtt{V_{max}}, \mathtt{V_{max}}] & \longrightarrow & [-\mathtt{N_q}, \mathtt{N_q}] \\ x & \longmapsto & \frac{x \times \mathtt{N_q}}{\mathtt{V_{max}}} \end{array} \tag{5.9}$$

où $\mathtt{N_q}$ est le nombre des niveaux de quantification et $\mathtt{V_{max}}$ est la borne supérieure positive des valeurs des caractéristiques. $\mathtt{V_{max}}$ peut être un hyper-paramètre tel que $\forall |x| > \mathtt{V_{max}}$, $x = \mathtt{V_{max}}$. $\mathtt{N_q}$ est un hyper-paramètre que nous avons déterminé empiriquement. La valeur $\mathtt{N_q} = 16$ donne les meilleures performances de reconnaissance.

Enfin, l'élaboration de notre système hybride avec un tel cadre applicatif permet de concevoir un système de reconnaissance robuste au manque de données d'apprentissage. Nous présentons dans la section suivante le protocole expérimental nous permettant d'évaluer les performances et la robustesse de notre système hybride.

5.4 Protocole expérimental

Dans cette section, nous explicitons notre protocole d'expérimentation. Nous commençons par présenter les bases de données exploitées, nous détaillons ensuite nos méthodes d'évaluation et nous présentons enfin les outils d'implémentation utilisés.

5.4.1 Bases de données

Les bases de données que nous avons exploitées pour évaluer notre système de reconnaissance et ses évolutions sont les bases de données proposées lors de la compétition `ChaLearn` 2011-2012 [54]. Nous n'avons pas participé à cette compétition mais nous avons pu comparer notre système avec les travaux des participants grâce à la plateforme de gestion des participants proposée pour cette compétition [4]. Nous détaillerons les résultats de ce classement dans la section 5.5.

Les bases de données `ChaLearn` présentent trois types de ressources : 480 sous-bases nommées `devel` à exploiter pendant la phase de développement du système de reconnaissance de gestes, 20 sous-bases nommées `valid` à exploiter pour valider le système et 40 sous-bases nommées `final` à exploiter pour l'évaluation finale du système. Les sous-bases `final` de 1 à 20 sont à tester au premier tour de la compétition et les bases de 21 à 40 sont à tester au deuxième tour de la compétition. Cette évaluation finale permet de classer les participants à la compétition `ChaLearn`.

Chacune de ces sous-bases contient 47 paires de vidéos. chaque paire de vidéos représente la même scène sous deux formats : format en couleur RGB et format en profondeur [5]. Ces vidéos sont enregistrées à l'aide d'une caméra Kinect (TM).

4. https ://www.kaggle.com/c/GestureChallenge2
5. Après normalisation, chaque pixel représente la distance entre la caméra et l'objet capturé.

Chaque vidéo contient 10 trames par seconde et chaque trame a une résolution de 240×320 pixels.

Les vidéos d'une même sous-base de données partagent les même caractéristiques scéniques : même acteur, même fond, mêmes conditions d'enregistrement et même thème et vocabulaire des gestes. 20 acteurs ont participé à l'élaboration de ces bases de données, un acteur par sous-base, et 30 vocabulaires de 8 à 15 gestes appartenant à des thèmes différents tels que les jeux vidéos, l'éducation à distance, le contrôle des robots, la langue des signes...

Chaque sous-base de données contient 100 gestes divisés en deux parties : un ensemble d'apprentissage $\mathbb{G}$ et un ensemble de test $\mathbb{S}$. L'ensemble d'apprentissage $\mathbb{G}$ est composé d'une dizaine de vidéos. Chaque vidéo contient une instance unique et isolée d'un geste qu'il faut identifier par la suite. Autrement dit, on ne dispose pour apprendre le modèle de chaque classe de geste que d'un seul exemple. Cette particularité est une contrainte supplémentaire de cette compétition puisqu'il faut pouvoir apprendre les modèles avec un seul exemple. D'où le thème de la compétition « one shot learning ».

L'ensemble de test $\mathbb{S}$ est composé d'une quarantaine de vidéos. Chaque vidéo contient une séquence de 1 à 5 gestes successifs séparés par un point de repos de l'ordre d'une seconde que nous appelons `transition`. La forme de cette `transition` est commune à toutes les séquences de test de toutes les bases. L'organisation des gestes dans les séquences de test est aléatoire, il n'y a pas d'ordre particulier. En d'autre termes, il n'existe pas une grammaire de gestes.

5.4.2 Métriques d'évaluation

Les organisateurs de la compétition `ChaLearn` ont défini une forme d'erreur globale sur l'ensemble des séquences de test basée sur la distance de `Levenshtein`[6], appelée également distance d'édition [78]. Cette forme d'erreur globale est notée par $\mathcal{L}_{\mathtt{ch}}$ et donnée par l'équation 5.10.

$$\begin{array}{rccc} \mathcal{L}_{\mathtt{ch}} : & \mathbb{D} & \longrightarrow & \mathbb{R} \\ & \mathbb{S} & \longmapsto & \frac{\sum_{s\in\mathbb{S}} \mathrm{L}(\mathcal{R}(s),\mathcal{T}(s))}{\sum_{s\in\mathbb{S}} l(\mathcal{T}(s))} \end{array} \tag{5.10}$$

où `s` est une séquence de gestes, $\mathcal{R}(s)$ est le résultat de reconnaissance du système pour la séquence `s`, $\mathcal{T}$ est une fonction donnant la séquence de gestes réelle correspondant à `s`, ce qu'on appelle la vérité terrain, $\mathrm{L}(.,.)$ est la distance de `Levenshtein`, $l(.)$ représente le nombre de gestes présents dans la séquence, $\mathbb{D}$ est l'ensemble des bases de données de test et $\mathbb{S}$ est un ensemble de séquences de test.

6. C'est une méthode de comparaison de séquences qui calcule le nombre minimal d'éléments qu'il faut supprimer, insérer ou remplacer pour passer d'une séquence à une autre.

Nous utilisons la forme de l'erreur $\mathcal{L}_{\texttt{ch}}$ pour nous situer par rapport aux travaux des participants à la compétition `ChaLearn`. Pour évaluer l'évolution de notre système et pour présenter les résultats marquants de nos différents tests, nous utilisons la forme classique de l'erreur calculée avec la distance de `Levenshtein` (équation 5.11). Cette forme d'erreur nous semble plus générique. Ainsi, pour calculer l'erreur globale sur un ensemble de séquence $\mathbb{S}$, nous calculons une moyenne que nous notons $\mathcal{L}$ donnée par l'équation 5.11.

$$\begin{array}{rccl} \mathcal{L}: & \mathbb{D} & \longrightarrow & [0,1] \\ & \mathbb{S} & \longmapsto & \frac{1}{|\mathbb{S}|}\sum_{s\in\mathbb{S}}\frac{\mathrm{L}(\mathcal{R}(s),\mathcal{T}(s))}{l(\mathcal{R}(s))+l(\mathcal{T}(s))} \end{array} \tag{5.11}$$

5.4.3 Outils d'implémentation

Nous avons exploité la bibliothèque OpenCV [19] pour les méthodes de traitement d'images et de vidéos. Pour la reconnaissance des gestes avec les `MMC`, nous avons exploité la bibliothèque Torch [7]. Pour la reconnaissance des gestes avec les `CAC`, nous avons ulitisé la bibliothèque CRF++ [8]. Toutes ces bibliothèques sont codée en C/C++.

5.5 Résultats de reconnaissance de gestes

Dans cette section nous présentons les résultats des différentes études réalisées avec les différentes variantes de nos systèmes de reconnaissance. Nous commençons par montrer l'intérêt de notre modèle de caractérisation du geste, la signature du geste. Ensuite, nous montrerons l'intérêt de la quantification des caractéristiques continues pour les `CAC` discrets. Puis, nous montrerons la robustesse des `CAC`/`MMC` à la variation du nombre du trames par état, à la variation de la durée des gestes et à la variation du vecteur de caractéristiques. Enfin, nous présenterons les résultats de reconnaissance de notre système hybride `CAC`/`MMC` comparés aux versions classiques et adaptées des `MMC` et des `CAC`. Nous terminons cette section, par la présentation de notre classement par rapport aux participants de la compétition `ChaLearn`.

Les résultats de reconnaissance présentés dans cette section correspondent à des tests de reconnaissance appliqués uniquement sur des données en format RGB sauf indication contraire. Nous indiquons dans tous les tableaux de résultats dans cette section le nombre de trames par état utilisé pour les `MMC`, noté $\texttt{f}_\texttt{e}$. Quand ce nombre change pour analyser son effet, l'étape d'apprentissage du système de reconnaissance est ré-appliquée. Tous les résultats de performances de reconnaissance présentés dans cette section correspondant au système hybride `CAC`/`MMC` sont obtenus à partir de tests réalisés avec un `CAC`/`MMC` adapté tel que expliqué dans la

7. http ://torch.ch/torch3/
8. http ://crfpp.googlecode.com/svn/trunk/doc/index.html

section 5.3, sauf indication contraire. Les systèmes de reconnaissance notés dans cette section par les « MMC adaptés » et les « CAC adaptés » sont également adaptés comme expliqué dans la section 5.3.

5.5.1 Intérêt de notre modèle de caractérisation de geste : la signature du geste

Dans cette sous-section, nous allons montrer l'intérêt de notre modèle de caractérisation du mouvement, la `signature` du geste, `SG`, à travers des résultats de reconnaissance.

Signature du geste et filtres particulaires. Le tableau 5.5 présente des résultats de reconnaissance avec deux modèles de caractérisation des gestes : une caractérisation avec la `signature` des gestes et une caractérisation avec la position de la main dominante calculée avec les filtres particulaires pour le même système de reconnaissance construit avec des modèles `MMC` adaptés. Ces résultats sont calculés sur deux bases `devel`. Ces résultats primaires montrent clairement qu'une caractérisation avec la signature des gestes permet d'obtenir de meilleures performances de reconnaissance par rapport à la caractérisation avec le suivi (tableau 5.5). De plus, en analysant les deux modèles de caractérisation, nous remarquons que la position de la main dominante est implicitement incluse dans la signature du geste. En effet, le terme de localisation du mouvement représente le centre d'inertie du mouvement, ainsi, si le mouvement est dominé par une main, il sera centré sur cette main. Contrairement au suivi des points d'intérêt comme la main dominante, la signature du geste, et en particulier le terme de localisation de mouvement, représente plusieurs cas de dominance de mouvement. Par exemple, si les deux mains effectuent deux mouvements de forces similaires, le terme de localisation de mouvement sera centré entre les deux mains, un cas de dominance que les filtres particulaires ne peuvent pas représenter. De plus, l'initialisation des filtres particulaires et la construction du vocabulaire de la référence nécessitent une connaissance *a priori* du type de la main dominante, gauche ou droite, or la méthode de `signature` du geste détecte automatiquement le point de dominanace. D'autre part, au niveau technique, les filtres particulaires nécessitent une étape supplémentaire de caractérisation de l'objet suivi, ce qui augmente la complexité du système de caractérisation. L'extraction de ces caractéristiques s'effectuent sur une centaine d'imagettes de particules pour chaque trame, or la signature du geste est calculée une seule fois par trame sur la totalité de l'image et ne nécessite pas une caractérisation particulière des objets mobiles. Pour ces raisons, la suite des expérimentations ne concerne que le système de caractérisation composé seulement de la `signature` du geste `SG` et les `HOG`.

Base de données	**Erreur : $\mathcal{L}$**		
	SG $l(\vec{c}) = 36$, $\mathtt{f_e} = 3$	SG $l(\vec{c}) = 18$, $\mathtt{f_e} = 4$	position de la main dominante $l(\vec{c}) = 4$
devel 01	0.07	0.10	0.46 ($\mathtt{f_e} = 14$)
devel 10	0.31	0.37	0.41 ($\mathtt{f_e} = 4$)

TABLE 5.5 – Les résultats de reconnaissance avec deux modèles de caractérisation des gestes : une caractérisation avec la signature des gestes et une caractérisation avec la position de la main dominante calculée avec les filtres particulaires pour le même système de reconnaissance, les MMC adaptés.

Signature du geste et descripteur HOG. Le tableau 5.6 présente les résultats de reconnaissance des deux systèmes, MMC adaptés et CAC/MMC, sur les bases devel avec trois variantes du vecteur de caractéristiques. Le but n'est pas de comparer ces deux systèmes de reconnaissance mais plutôt de valider l'intérêt du vecteur de descripteurs SG. Selon le tableau 5.6, nous constatons que les performances des systèmes de reconnaissance avec le vecteur de descripteurs SG appliqué sur les données au format RGB et au format profondeur, sont très proches des performances de ces systèmes de reconnaissance utilisant un vecteur de caractéristiques qui combine le vecteur SG et le vecteur HOG. De plus, ces valeurs de l'erreur sont faibles et présentent des performances de reconnaissance intéressantes. Ainsi, appliqué sur les deux formats de données, la signature du geste SG peut représenter un modèle de caractérisation complet. Ce que nous pouvons constater également à partir du tableau 5.6 c'est que les performances des CAC/MMC ont diminué quand la modalité de profondeur a été ajouté. Cela donne une première indication que les CAC/MMC ne sont pas toujours sensibles à l'augmentation du vecteur des caractéristiques. Nous détaillerons davantage cette constatation dans la sous-section 5.5.3.

Système	$\mathtt{f_e}$	**Erreur : $\mathcal{L}$**		
		format RGB	format RGB & profondeur	
		(SG,HOG) $l(\vec{c}) = 180$	(SG,HOG) $l(\vec{c}) = 360$	SG $l(\vec{c}) = 72$
MMC adaptés	3	0.21	0.20	0.21
MMC adaptés	5	0.25	0.24	0.25
CAC/MMC	5	0.22	0.22	0.26

TABLE 5.6 – Les résultats de reconnaissance avec différentes variantes du vecteur de caractéristiques sur les 20 bases devel (en total 750 séquences de test de l'ordre de 200 trames chacune).

5.5.2 Intérêt de la quantification des caractéristiques continues pour les CAC

Nous commençons par montrer l'intérêt de la quantification des valeurs des caractéristiques pour les CAC. La figure 5.8 présente les performances de reconnaissance du CAC/MMC à caractéristiques continues et discrètes en variant le nombre de trames par état pour la composante MMC. Puisque le CAC représente la composante qui gère les caractéristiques pour le CAC/MMC, le type de caractéristiques discret ou continu concerne particulièrement les CAC. Les performances du système discret dépassent largement les performances du système continu ce qui prouve l'intérêt de la quantification. L'erreur élevée des CAC continus dans ce cas provient de l'incohérence entre les probabilités des classes données par les CAC et les probabilités de transitions données par les MMC. Par exemple, pendant le décodage, l'algorithme de Viterbi peut commencer un chemin en affectant un premier état d'un geste puis ne trouve pas la solution souhaitée pour le reste des états de ce geste.

Par ailleurs, la durée d'apprentissage des CAC continus (estimée en heures) dépasse largement la durée d'apprentissage des CAC discrets (estimée en minutes). Cela présente un autre avantage pour les CAC discrets.

Une autre différence entre les CAC continus et les CAC discrets appuyant l'intérêt des CAC discrets est la méthode de gestion des poids des fonctions caractéristiques. Les CAC continus réservent un seul poids pour toutes les valeurs d'un descripteur. Or une valeur réduite de ce descripteur ne signifie pas forcément qu'elle n'a pas d'intérêt et une valeur élevée de ce descripteur ne signifie pas aussi qu'elle est intéressante. Autrement dit, l'algorithme d'apprentissage des CAC continus donne la même pondération pour toutes les valeurs d'un même descripteur. Cette façon de gérer les poids peut être adéquate pour pondérer une fonction score dont l'intérêt des valeurs est monotone. Par contre, pour un descripteur, le caractère distinctif des plages de valeurs peut changer d'un descripteur à l'autre. Ainsi, les CAC discrets, qui donnent un poids par valeur discrête des caractéristiques, spécialisent davantage les caractéristiques, ce qui augmente par la suite la discrimination des classes. Pour cette raison, les CAC discrets présentent un modèle adéquat pour le cas d'apprentissage avec un seul exemple comme nous l'avons indiqué dans la sous-section 5.3.2.

5.5.3 Robustesse des CAC/MMC

5.5.3.1 Robustesse à la variation du nombre de trames par état

Les CAC/MMC sont robustes à la variation du nombre de trames par état. En effet, la figure 5.9 présente la variation de l'erreur de reconnaissance $\mathcal{L}$ des systèmes MMC adaptés et CAC/MMC en fonction du nombre de trame par état $\mathtt{f_e}$ (ce paramètre influence principalement la composante MMC). Pour chaque nouvelle valeur de $\mathtt{f_e}$, le système de reconnaissance est ré-appris. Nous constatons dans la figure 5.9 que

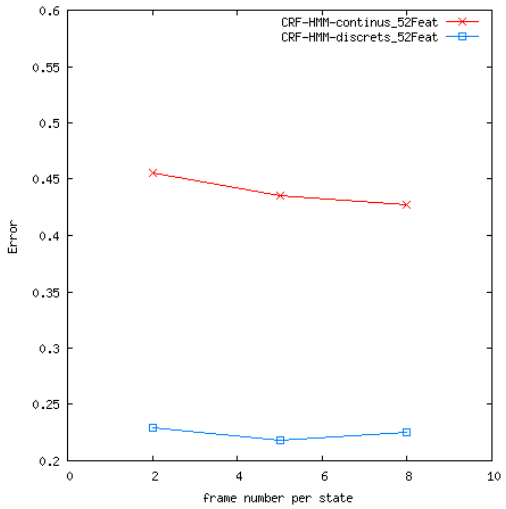


FIGURE 5.8 – Les résultats de la reconnaissance des gestes avec les CAC/MMC à composante CAC continue et discrète sur les 20 bases devel (en total 750 séquences de test de l'ordre de 200 trames chacune)

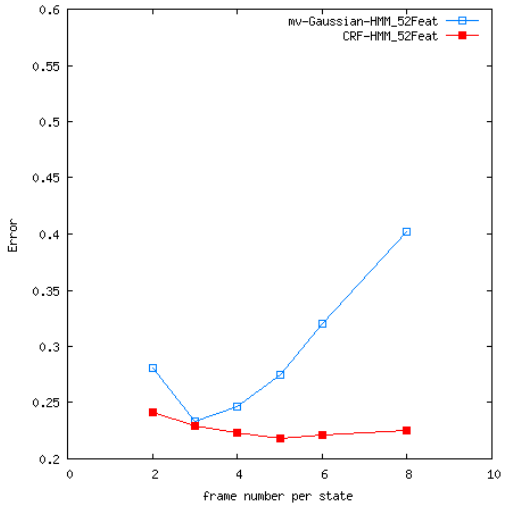
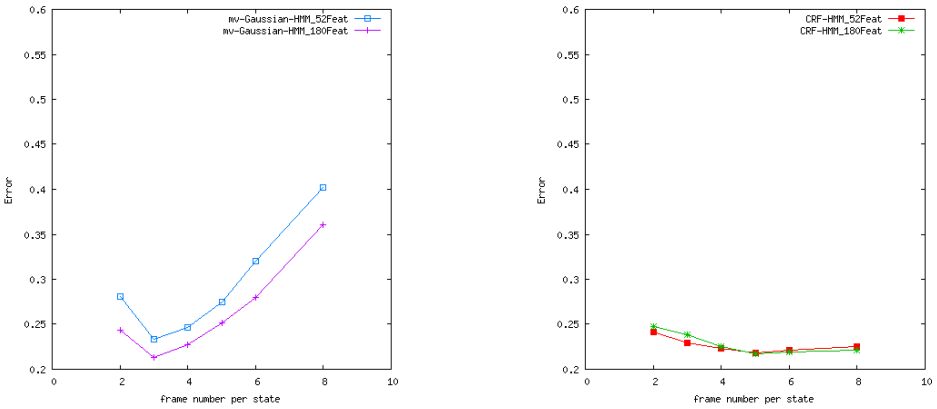


FIGURE 5.9 – Robustesse du système CAC/MMC à la variation du nombre de trames par états comparée aux MMC adaptés (performances sur les 20 bases devel (en total 750 séquences de test de l'ordre de 200 trames chacune))



(a) les MMC adaptés (b) les CAC/MMC

FIGURE 5.10 – Robustesse à la variation du vecteur des caractéristiques (performances sur les 20 bases `devel` (en total 750 séquences de test de l'ordre de 200 trames chacune)).

l'erreur de reconnaissance des CAC/MMC est faible et beaucoup plus stable pour différentes valeurs du nombre de trames. Ils sont capables de gérer un nombre élevé de trames par état. Cependant, les MMC classiques ou adaptés sont très sensibles à la variation du nombre de trames par état. Ils présentent une erreur très élevée dans le cas d'un grand nombre de trames par état comme le montre la figure 5.9.

5.5.3.2 Robustesse à la variation de la durée du geste

La variation du nombre de trames par état a une conséquence directe sur la robustesse à la variation de la durée du geste. Avec un grand nombre de trames par état, le système devient capable de gérer l'élasticité temporelle du geste. Autrement dit, quand le geste se dilate ou se rétrécie en nombre de trames dans les données de test, le système reste capable d'aligner le modèle sur les données et les décoder. De plus, les CAC sont capables de gérer implicitement le rétrécissement et la dilatation des données grâce à leur décision locale indépendante du modèle global des données, contrairement aux MMC qui dépendent d'un modèle graphique orienté sans sauts. Ainsi, pour gérer l'élasticité temporelle des gestes, une structure simple du modèle hybride avec un nombre réduit d'états peut remplacer un système MMC complexe avec des sauts entre les états et une connexion complète comme adoptés par certains participants de la compétition ChaLearn [73, 62, 146].

5.5.3.3 Robustesse à la variation du vecteur des caractéristiques

Les figures 5.10 présentent la variation de l'erreur $\mathcal{L}$ en fonction du nombre de trames par état $\mathtt{f_e}$ pour deux systèmes MMC (figure 5.10a) et pour deux systèmes CAC/MMC (figure 5.10b). Chaque paire de systèmes est évaluée avec deux vecteurs

de caractéristiques différents. La différence est au niveau de la précision de la description. Les régions de l'image où les caractéristiques du vecteur $l(\vec{c}) = 52$ sont calculées, sont encore divisées pour le vecteur des caractéristiques $l(\vec{c}) = 180$. Quand la taille du vecteur des caractéristiques diminue, les CAC/MMC gardent quasiment les mêmes performances. Autrement dit, un minimum de caractéristiques suffit pour les CAC/MMC, alors que pour les MMC classique l'ajout des caractéristiques augmente considérablement les performances de reconnaissance. Cette capacité de reconnaissance avec un nombre de caractéristiques réduit rend la tâche d'extraction des caractéristiques plus simple et plus rapide. En traitement de vidéos/images, cette tâche demande en général beaucoup de temps et de ressources.

Ces trois caractères de robustesse des CAC/MMC montrent qu'avec un système simple, un niveau élevé de performance de reconnaissance est possible grâce à la combinaison des avantages et la compensation des inconvénients des CAC et des MMC. Nous pouvons constater la simplicité du système des CAC/MMC sur trois niveaux : (a) une structure simple du modèle avec un nombre d'état réduit (signifie un nombre de trames par état élevé, meilleurs performances pour $\mathtt{f_e} = 5$) sans sauts et sans connexion complète. (b) un nombre réduit de caractéristiques. (c) un jeu de données d'apprentissage réduit à un exemple par classe.

5.5.4 Validation du système hybride CAC/MMC et son évaluation avec la plateforme ChaLearn

Nous présentons dans cette sous-section les résultats de reconnaissances de notre système hybride CAC/MMC sur les bases valid et les bases final, ce qui va nous permettre de le positionner dans le classement des systèmes de reconnaissance des participants à la compétition ChaLearn.

Nous commençons d'abord par présenter des résultats de reconnaissance sur les bases devel en comparant les performances des principaux systèmes de reconnaissance que nous avons étudiés (tableau 5.7). Le vecteur des caractéristiques est identique pour tous ces systèmes et le nombre de trames par état $\mathtt{f_e}$ correspond à la valeur donnant le meilleur résultat de reconnaissance pour chaque système. $\mathtt{f_g}(\mathtt{g})$ représente la taille du geste appris, ce qui signifie que chaque geste est représenté par une seule classes, les sous-classes qui correspondent aux états dans le cas des MMC n'existent pas dans ce cas. D'autre part, une étape de post-traitement est appliquée sur les CAC classique et adapté afin de filtrer leurs résultats de reconnaissance, sans cette étape l'erreur de reconnaissance dépasse 0.5. Le tableau 5.7 montre clairement que les performances de reconnaissance du système hybride proposé CAC/MMC dépassent les performances de reconnaissance des autres systèmes classiques ou adaptés.

Pour valider et classer les CAC/MMC par rapport aux travaux des participants à la compétition ChaLearn 20111-2012, nous avons testé ce système hybride sur les bases valid, final 1-20 et final 21-40 fournies dans le cadre de cette compéti-

Système	$l(\vec{c})$	$\mathtt{f_e}$	**Erreur : $\mathcal{L}$**
MMC classique	52	6	0.36
MMC adapté	52	3	0.23
CAC classique (continu)	52	$\mathtt{f_g}(\mathtt{g})$	0.29
CAC adapté (discret)	52	$\mathtt{f_g}(\mathtt{g})$	0.28
CAC/MMC (adapté)	52	5	**0.22**

TABLE 5.7 – Les résultats de reconnaissance de différents systèmes de reconnaissances basé sur les MMC et les CAC sur les 20 bases devel (en total 750 séquences de test de l'ordre de 200 trames chacune)

tion. Le tableau 5.8 présente les valeurs de l'erreur de reconnaissance du système hybride CAC/MMC sur les bases valid, final 1-20 et final 21-40 selon les deux méthodes d'évaluation $\mathcal{L}$ (équation 5.11) et $\mathcal{L}_{\mathtt{ch}}$ (équation 5.10). Le tableau 5.8 présente également le classement des performance du système CAC/MMC par rapport aux travaux des participants à la compétition sur les trois catégories de bases de données. Ce classement est obtenu en utilisant la valeur de l'erreur $\mathcal{L}_{\mathtt{ch}}$. Nous avons deux références pour obtenir ce classement : la liste des scores (correspondant à l'erreur $\mathcal{L}_{\mathtt{ch}}$) de tous les participants fournie par les organisateurs de la compétition pour le premier tour [9] et pour le deuxième tour [10], et la plateforme d'évaluation de la compétition ChaLearn [11]. La liste complète des scores pour les bases valid n'est pas accessible, ainsi nous ne pouvons pas donner notre classement sur ces bases. L'évaluation des performances de reconnaissance d'un système nécessite la soumission des résultats de reconnaissance détaillées sur chaque catégorie de bases de données et pour chaque séquence de test, mais le résultat final fourni est le classement selon les bases de données du deuxième tour seulement de la compétition. Ainsi, pour classer les performances du système hybride CAC/MMC sur les bases final du premier tour, nous nous sommes référés à la liste des scores présentée dans la figure 5.11, et pour classer les performances du CAC/MMC sur les bases final du deuxième tour, nous nous sommes référés à la plateforme de la compétition qui nous a retourné le résultat présenté dans la figure 5.12. Les deux figures 5.11 et 5.12 sont des copies des pages web correspondant à la compétition ChaLearn gérées par le site des compétitions « *Kaggle* ».

Le tableau 5.8 présente les résultats les plus importants pour nos travaux : les résultats de reconnaissance de notre système hybride CAC/MMC sur les bases final 21-40 du premier tour de la compétition ChaLearn 2011 et sur les bases final 21-40 du deuxième et dernier tour de la compétition ChaLearn 2012. Avec ces résultats nous avons été classés en $7^{\mathbf{ème}}$ **position** sur la liste des participants au premier tour de la compétition (figure 5.11) et en $7^{\mathbf{ème}}$ **position** également sur

9. https ://www.kaggle.com/c/GestureChallenge/leaderboard
10. https ://www.kaggle.com/c/GestureChallenge2/leaderboard
11. http ://www.kaggle.com/c/GestureChallenge/details/submission-instructions

Base de données	Erreur		classement
	$\mathcal{L}$	$\mathcal{L}_{\mathtt{ch}}$	
`valid`	0.177193	0.348812	-
`final` 1-20 (1^{er} tour)	0.147924	0.296440	$7^{\text{ème}}$
`final` 21-40 ($2^{\text{ème}}$ tour)	0.122398	0.252357	$7^{\text{ème}}$

TABLE 5.8 – Les résultats de reconnaissance du système hybride `CAC/MMC` sur les 20 bases `valid`, 20 bases `final` 1-20 et 20 bases `final` 21-40 avec $l(\vec{c}) = 180$ et $\mathtt{f_e} = 5$ (chaque catégorie de bases de données contient en total environ 750 séquences de test de l'ordre de 200 trames chacune)

la liste des participants au deuxième de tour de cette compétition (figure 5.12). Nous avons réussi à atteindre ce classement en utilisant uniquement les données en format RGB.

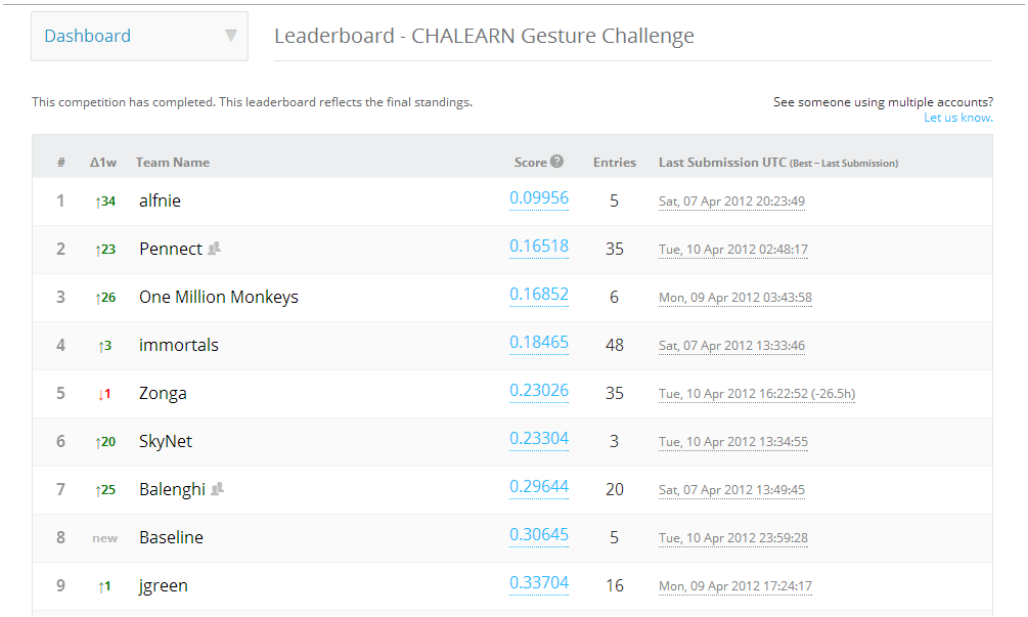


FIGURE 5.11 – Les scores des 9 premiers travaux des participants à `ChaLearn` 2011, 1^{er} tour (les performances de notre système `CAC/MMC` (tableau 5.8) sont ex æquo avec le score de la $7^{\mathbf{ème}}$ **position**).

Ces résultats et cette étude montrent que le système hybride `CAC/MMC`est un système qui présente de meilleurs performances par rapport à d'autres systèmes classiques, robuste à différentes variations et intéressant dans le cas réel avec peu données d'apprentissage. Pour valider davantage notre système hybride `CAC/MMC`, nous avons effectué un ensemble de tests statistiques. Nous présentons cette vérification statistique dans la section suivante.

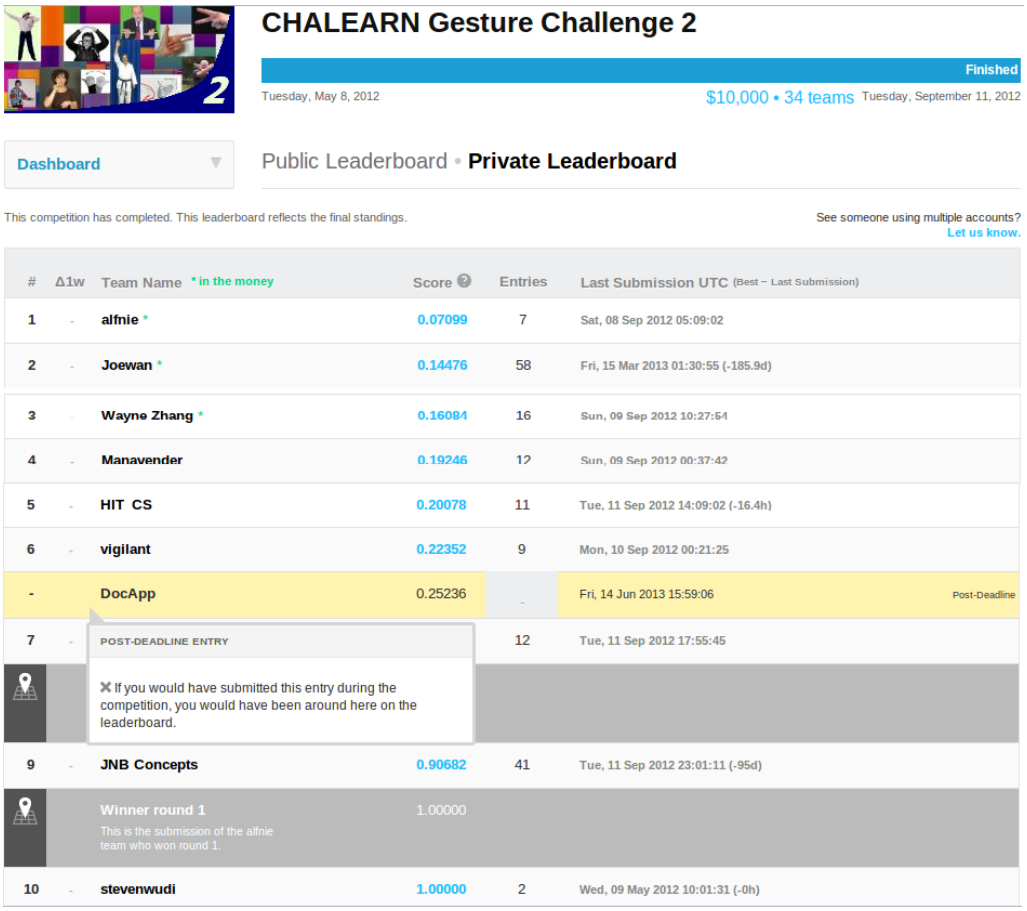


FIGURE 5.12 – Position de notre système hybride (indexé par le nom de notre équipe « *DocApp* ») dans le classement final des performances des participants à `ChaLearn` 2012, 2ème tour : **7ème position** parmi les dix premiers (post-soumission).

5.6 Vérifications statistiques

Nous avons effectué un ensemble de tests statistiques pour montrer l'intérêt de l'adaptation au cas d'apprentissage avec un seul exemple que nous avons apporté aux `MMC` et pour valider davantage l'apport de notre modèle hybride `CAC`/`MMC`.

5.6.1 Le test de `Kolmogorov-Smirnov`

Ayant un seul exemple d'apprentissage, il est difficile de générer un modèle générique représentatif des données. Néanmoins, avec nos expériences nous avons montré qu'avec une adaptation du modèle, il est possible de le généraliser. Le test de `Kolmogorov-Smirnov` peut nous aider à vérifier cette constatation. Ce test statistique permet d'évaluer la conformité entre deux distributions.

Pour vérifier si un modèle Markovien à temps discret est bien représentatif des données, il faut que le temps de séjour $\mathtt{T}$ dans un état $\mathtt{e}_i$ suive une loi géométrique de paramètre $1 - p(y_t = \mathtt{e}_i | y_{t-1} = \mathtt{e}_i)$. Ce paramètre peut être estimé au cours du décodage de toutes les séquences de test tel que $p(y_t = \mathtt{e}_i | y_{t-1} = \mathtt{e}_i) = \frac{\text{nombre de transition de } \mathtt{e}_i \text{ vers } \mathtt{e}_i}{\text{nombre total de visite de } \mathtt{e}_i}$. Il est possible également d'estimer la probabilité de temps de séjour $\mathtt{T}_{\mathtt{e}_i}$ pour toutes les séquences de test tel que $p(\mathtt{T}_{\mathtt{e}_i} = k) = \frac{\text{nombre de visite de } \mathtt{e}_i \text{ avec } \mathtt{T}_{\mathtt{e}_i}=k}{\text{nombre total de visite de } \mathtt{e}_i}$. Par la suite, le test de `Kolmogorov-Smirnov` permet de calculer le degré de conformité, `p-value`, entre la distribution du temps de séjour dans un état $\mathtt{e}_i$ calculée expérimentalement et sa distribution théorique selon le modèle Markovien. Si la `p-value` est supérieur à 0.05 alors l'hypothèse de conformité est vérifiée.

Pour réaliser ce test, nous avons choisi l'état $\hat{\mathtt{e}}_i$ ayant le nombre le plus élevé de visites pendant l'étape de décodage. Nous avons appliqué ce test avec différentes configurations du système `MMC`. D'abord, nous avons conçus plusieurs versions du vecteur de caractéristiques basées sur la signature du geste extraite de la vitesse des flots optiques et sur les informations extraites des `HOG` expliquées dans la sous-section 5.2.3. Ensuite, quand la configuration du vecteur de caractéristiques change, il est nécessaire de changer le nombre de trames par état $\mathtt{f}_\mathtt{e}$ pour avoir de meilleurs performances de reconnaissance. Pour chaque configuration du vecteur de caractéristiques, nous avons appliqué le test de `Kolmogorov-Smirnov` sans et avec adaptation de la variance des gaussiennes.

La figure 5.13 représente le résultat du test de `Kolmogorov-Smirnov` pour un système `MMC` avec 6 configurations différentes du vecteur de caractéristiques sans et avec adaptation des gaussiennes. Les 6 configurations sans adaptation sont représentées sur la figure par les abscisses de 1 à 6, et les mêmes configurations avec adaptation sont représentées par les abscisses de 7 à 12 selon le même ordre. Nous remarquons que la `p-value` des configurations sans adaptation est presque nulle, tandis que la `p-value` des configurations avec adaptation s'approche du seuil 0.05 et le dépasse pour certaines configurations, ce qui montre que l'hypothèse de conformité est bien vérifiée pour ces dernières configurations. Ces dernières configurations représentent des vecteur de caractéristiques basés que sur les informations des flots optiques. Nous rappelons que quand la `p-value`est supérieur à 0.05, la distribution du temps de séjour à laquelle appartient l'échantillon testé est conforme à une lois géométrique, ce qui signifie que le modèle `MMC` testé est bien un modèle générique. Ainsi, le fait d'unifier les variances des gaussiennes et de les remplacer par une moyenne des variances rend bien le modèle `MMC` adapté au cadre d'apprentissage avec un seul exemple.

5.6.2 Le test de `Student` unilatéral

Afin de confirmer que les performances de reconnaissance de notre système hybride dépassent significativement les performances des `MMC` et des `CAC` clas-

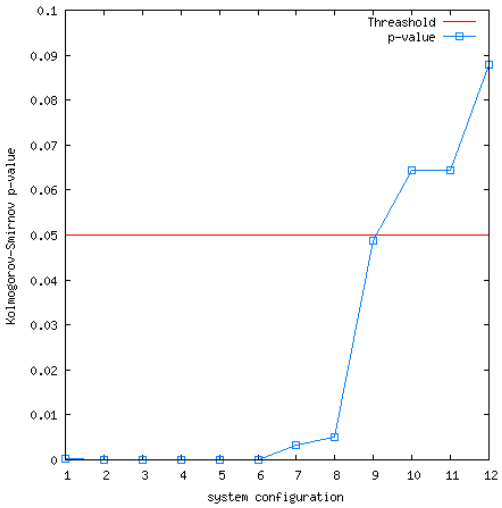


FIGURE 5.13 – La `p-value`du test de `Kolmogorov-Smirnov` calculé pour l'état le plus visité pour différentes configurations du modèle `MMC`

siques, nous comparons les erreurs globales associées $\mathcal{L}$ (équation 5.11). $\mathcal{L}$ est une moyenne de l'erreur calculée sur l'ensemble des 20 bases de données `devel` contenant en total 750 séquences de test. Nous calculons également la variance associée $\sigma_{\mathcal{L}}$ et nous appliquons la formule 5.12 du test de `Student` pour chaque couple de systèmes testés. En cherchant dans la table de la loi de Student, la valeur $t_s(\mathbb{D})$ doit dépasser une valeur t_α à laquelle correspond une erreur α. Ayant cette condition, le niveau de confiance obtenu selon un test unilatéral appuie l'hypothèse que la moyenne d'erreur $\mathcal{L}_1$ est significativement différente de la moyenne d'erreur $\mathcal{L}_2$.

$$t_s(\mathbb{D}) = \frac{|\mathcal{L}_1(\mathbb{D}) - \mathcal{L}_2(\mathbb{D})|}{\sqrt{\sigma^2_{\mathcal{L}_1}(\mathbb{D}) + \sigma^2_{\mathcal{L}_2}(\mathbb{D})}} \sqrt{|\mathbb{D}| - 1} \tag{5.12}$$

Pour une taille de données égale à 750, nous montrons avec le test de `Student` unilatéral que les performances de reconnaissance de notre modèle hybride `CAC`/`MMC` dépassent significativement les performances de reconnaissance des deux modèles classiques `MMC` et `CAC`. Les performances des `CAC`/`MMC` dépassent les performances des `MMC` adaptés [12] avec un niveau de confiance de 99% ($t_s(\mathbb{D}) = 2,525001$) et dépassent les performances des `CAC` adaptés [12] avec un niveau de confiance de 99.5% ($t_s(\mathbb{D}) = 4,456314$).

12. L'adaptation mentionnée est l'adaptation du modèle au cas de l'apprentissage avec un seul exemple.

Conclusion

Dans ce chapitre, nous avons présenté un nouveau système hybride de reconnaissance des gestes CAC/MMC. Nous avons montré que cette combinaison de modèles markoviens renforce les avantages de chaque modèle et compense leurs inconvénients. Le cadre applicatif de nos systèmes de reconnaissance est l'apprentissage avec un seul exemple. Nous avons appliqué des méthodes d'adaptation de ces systèmes markoviens pour renforcer leur capacité de reconnaissance avec peu de données d'apprentissage. Nous avons présenté également un nouveau modèle de caractérisation des gestes qui est une `signature` de geste fondée sur les flots optiques.

Nous avons montré que ces modèles de caractérisation et de reconnaissance des gestes forment un système hybride robuste de reconnaissance qui ouvre de nouvelles perspectives pour les modèles markoviens séquentiels.

Conclusion Générale

L'analyse des gestes est un axe de recherche lié à plusieurs domaines d'interaction Homme-Machine comme la traduction des langues gestuelles, les commandes se robots et les jeux vidéos. Les techniques d'analyse des gestes sont applicables également pour la gestion des documents vidéos. C'est dans ce cadre que nous avons effectué notre étude. Les types de gestes étudiés ne sont pas limités à une catégorie précise. La problématiques que nous avons posée au début de cette thèse était la modélisation spatiotemporelle des données séquentielles, notamment les gestes.

Pour répondre à cette problématique, nous avons élaboré un modèle hybride markovien `CAC/MMC` pour la reconnaissance de gestes. Ce système hybride combine un modèle discriminant les données de type champ aléatoire conditionnel et un modèle génératif de type `MMC` qui permet de contrôler le processus de reconnaissance par un modèle probabiliste agissant comme un modèle de langage. Le principe de ce modèle est d'intégrer les probabilités à posteriori, calculées par les `CAC`, dans le processus de décodage des `MMC` en remplaçant le modèle d'attache aux données gaussien. Ce modèle hybride combine la capacité de segmentation et de structuration globale des `MMC` et la capacité de discrimination locale des `CAC`.

Nous avons testé nos systèmes sur les données de la compétition `ChaLearn` 2011-2012 « *Gesture Challenge 1-2* » qui a pour contrainte supplémentaire de ne fournir qu'un seul exemple d'apprentissage de chaque classe de geste. Nous avons conçu ainsi une méthode d'adaptation des `MMC` et des `CAC` à ce cadre applicatif. Nous avons adapté les `MMC` à travers l'unification des valeurs des variances du modèle d'attache aux données gaussien. Les `CAC` ont été adapté avec la quantification des valeurs continues des caractéristiques observées.

Nous avons montré à travers nos expérimentations l'efficacité de ces adaptations. Nous avons montré également l'efficacité du modèle hybride proposé, les `CAC/MMC`. Ses performances de reconnaissance dépassent les performances de reconnaissance des `MMC` et des `CAC` classiques ou adaptés. De plus, nous avons montré que les `CAC/MMC` sont robustes aux variations du nombre d'états correspondant aux sous-classes des gestes, à la variation de la durée du geste, à la variation du vec-

teur de caractéristiques et au manque des données d'apprentissage. Le système d'évaluation de la compétition `ChaLearn` a classé les résultats de reconnaissance de notre système hybride en $7^{\text{ème}}$ position. Nous avons réussi à atteindre ce classement en utilisant uniquement les données en format RGB.

Nous avons conçu un modèle de caractérisation du mouvement que nous avons appelé la `signature` du geste. Ce modèle se compose de trois catégories de caractéristiques : des caractéristiques de localisation, des caractéristiques d'orientation, et des caractéristiques de vélocité de mouvement. Il est construit à partir des informations de vitesse extraites avec la technique des flots optiques. À travers nos expérimentations, nous avons montré que ce modèle de caractérisation est capable de fournir une description complète, discriminante, de faible complexité, et robuste à des variations de couleur, de forme, de fond, etc. L'ajout d'autres descripteurs à ce modèle de caractérisation, comme les `HOG`, a une influence très faible sur les résultats de reconnaissances. De plus, nous avons montré que l'information de la position de la main dominante fournie par les filtres particulaires est incluse implicitement dans le modèle de signature du geste, en particulier dans le sous-modèle de localisation du mouvement. En comparant le modèle de suivi de la main dominante et le modèle de la signature du geste sur plusieurs niveaux tels que les performances de reconnaissance, la complexité, l'invariance, la précision et la variété d'informations fournies, nous avons conclu que la signature du geste s'avère plus adéquate pour la caractérisation des gestes pour la tâche de reconnaissance envisagée.

Néanmoins, nos contributions réalisées pour le suivi des geste restent applicables dans d'autres domaines de suivi utilisant les filtres particulaires comme la vidéo-surveillance, la navigation des robots, le contrôle du trafic des voitures, etc. Notre contribution principale dans ce contexte a été l'élaboration d'un modèle de pénalisation des particules basé sur les flots optiques afin d'améliorer le suivi. Nous avons également proposé une méthode de construction automatique d'un vocabulaire de référence de la cible suivi, sans avoir recours à une phase d'apprentissage. À travers les expérimentation réalisées, nous avons montré que les contributions apportées aux filtres particulaires améliorent la qualité du suivi par rapport aux filtres particulaires classiques et permettent au filtre de suivre les mouvements rapides et irréguliers de la cible. Afin d'éliminer le risque de confusion dans le cas de présence de différents objets similaires mobiles dans la scène, il est s'avère nécessaire d'appliquer une méthode de suivi multiple. Il existe des variantes des filtres particulaires pour le suivi multiple et leur étude représente une perspective intéressante de notre travail de suivi des gestes.

Une deuxième perspective intéressante dans la continuation de notre travail de reconnaissance de gestes concerne la tâche de détection de gestes, ce qu'on appelle le « spotting » de gestes. La détection des gestes s'applique dans les contextes de gestion des documents vidéo tels que la recherche des vidéos et la catégorisation

et l'indexation des vidéos. La tâche de détection des gestes consiste à localiser et étiqueter des gestes spécifiques dans des vidéos quelconques contenant d'autres informations que l'information recherchée. Notre modèle de reconnaissance serait applicable dans ce cas en représentant les faux exemples par une classe complémentaire au vocabulaire des gestes à détecter.

Enfin, nous avons montré dans cette thèse les capacités des systèmes markoviens à modéliser et gérer les variations spatiotemporelles des données séquentielles, notamment les gestes. L'évolution de la modélisation de l'activité humaine participe à l'évolution des techniques de vision par ordinateur et par la suite participe à l'évolution des systèmes d'interaction Homme-Machine.

Bibliographie

[1] *Joint probabilistic techniques for tracking multi-part objects*, 1998. [cited at p. 57]

[2] S Adam, J-M Ogier, C Cariou, R Mullot, J Gardes et Y Lecourtier : Utilisation de la transformée de fourrier-mellin pour la reconnaissance de formes multi-orientées et multi-échelles : application à l'analyse automatique des documents techniques. *Revue Traitement du Signal*, 18(1):17–33, 2001. [cited at p. 28]

[3] K. Ait-Mohand : *Techniques d'adaptation de modèles markoviens. Application à la reconnaissance de documents anciens.* Thèse de doctorat, 2011. [cited at p. 66]

[4] Mark A. Aizerman : Review of "syntactic methods in pattern recognition" by king-sun fu. *IEEE Transactions on Systems, Man, and Cybernetics*, 6(8):590–591, 1976. [cited at p. 27]

[5] M. AL-Rousan, K. Assaleh et A. Tala'a : Video-based signer-independent arabic sign language recognition using hidden markov models. *Applied Soft Computing*, 9(3):990 – 999, 2009. [cited at p. 68]

[6] Elise Arnaud : *Methodes de filtrage pour du suivi dans des sequences d'images - Application au suivi de points caracteristiques.* These, Université Rennes 1, novembre 2004. [cited at p. 12]

[7] M.S. Arulampalam, S. Maskell, N. Gordon et T. Clapp : A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Trans. Sig. Proc.*, 50(2):174–188, feb 2002. [cited at p. 12]

[8] Marcell Assan et Kirsti Grobel : Video-based sign language recognition using hidden markov models. *In Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction*, pages 97–109, London, UK, 1998. Springer-Verlag. [cited at p. 68]

[9] S. Austin, R. Schwartz et P. Placeway : The forward-backward search algorithm. *In Proceedings of the Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference*, ICASSP '91, pages 697–700, Washington, DC, USA, 1991. IEEE Computer Society. [cited at p. 85]

[10] E. Barbu : *Fouille et classification de graphes. Application à la reconnaissance de symboles dans les documents techniques.* Thèse de doctorat, 2007. [cited at p. 28]

[11] F. Bashir et F. Porikli : Performance evaluation of object detection and tracking systems. *In IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, jun 2006. [cited at p. 31, 32, 49, 57]

[12] L. E. Baum et T. Petrie : Statistical inference for probabilistic functions of finite state Markov chains. *Annals of Mathematical Statistics*, 37:1554–1563, 1966. [cited at p. 63]

[13] Yoshua Bengio, Yann LeCun, Craig Nohl et Chris Burges : Lerec : A nn/hmm hybrid for on-line handwriting recognition. *Neural Comput.*, 7(6):1289–1303, novembre 1995. [cited at p. 78]

[14] Donald J. Berndt et James Clifford : Using Dynamic Time Warping to Find Patterns in Time Series. *In KDD Workshop*, pages 359–370, 1994. [cited at p. 62]

[15] Suchendra M. Bhandarkar et Xingzhi Luo : Integrated detection and tracking of multiple faces using particle filtering and optical flow-based elastic matching. *CVIU*, 113(6):708–725, June 2009. [cited at p. 7, 24, 25, 30, 45, 89]

[16] A. Bhattacharyya : On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943. [cited at p. 44]

[17] Ye Bin et Peng Jia-xiong : Improvement and invariance analysis of zernike moments using as a region-based shape descriptor. *In Proceedings of the 15th Brazilian Symposium on Computer Graphics and Image Processing*, SIBGRAPI '02, pages 120–, Washington, DC, USA, 2002. IEEE Computer Society. [cited at p. 29]

[18] Alan C. Bovik : Analysis of multichannel narrow-band filters for image texture segmentation. *IEEE Transactions on Signal Processing*, 39(9):2025–2043, 1991. [cited at p. 27]

[19] Gary Bradski et Adrian Kaehler : *Learning OpenCV : Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA, 2008. [cited at p. 7, 15, 23, 30, 36, 44, 100]

[20] Gary R. Bradski : Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, (Q2), 1998. [cited at p. 132]

[21] Annelies Braffort : Argo : An architecture for sign language recognition and interpretation. *In Proceedings of Gesture Workshop on Progress in Gestural Interaction*, pages 17–30, London, UK, 1997. Springer-Verlag. [cited at p. 34]

[22] Lars Bretzner, Ivan Laptev et Tony Lindeberg : Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering. *In Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 423–, Washington, DC, USA, 2002. IEEE Computer Society. [cited at p. 24]

[23] R. Brunelli, O. Mich et C. M. Modena : A survey on video indexing. *Journal of Visual Communication and Image Representation*, 10:78–112, 1996. [cited at p. 4]

[24] A. Bugeau et P. Pérez : Detection and segmentation of moving objects in highly dynamic scenes. *In Proc. Conf. Computer Vision and Pattern Recog. (CVPR' 07)*, pages 1–8, Minneapolis, MI, June 2007. [cited at p. 12]

[25] Aurélie Bugeau : These, Université Rennes 1, Dec 2007. [cited at p. 29]

[26] J Canny : A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, juin 1986. [cited at p. 28]

[27] Vicent Caselles, Francine Catté, Tomeu Coll et Françoise Dibos : A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31, dec 1993. [cited at p. 28]

[28] Seong-Sik Cho, Hee-Deok Yang et Seong-Whan Lee : Sign language spotting based on semi-markov conditional random field. *Workshop on Applications of Computer Vision (WACV)*, pages 1 – 6, December 2009. [cited at p. 77]

[29] Jae Gark Choia et Seong-Dae Kim : Multi-stage segmentation of optical flow field. *Signal Processing*, 54(2):109 – 118, 1996. [cited at p. 29]

[30] Chee-Way Chong, P Raveendran et R Mukundan : Translation invariants of zernike moments. *Pattern Recognition*, 36(8):1765–1773, 2003. [cited at p. 29]

[31] Luigi Cinque, Stefano Levialdi, Kai A. Olsen et A. Pellicanò : Color-based image retrieval using spatial-chromatic histograms. *In ICMCS, Vol. 2*, pages 969–973, 1999. [cited at p. 27]

[32] D. Comaniciu, V. Ramesh et P. Meer : Real-time tracking of non-rigid objects using mean shift. *In Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 142–149. IEEE, 2000. [cited at p. 12, 27]

[33] Andrea Corradini : Real-time gesture recognition by means of hybrid recognizers. *In Gesture Workshop*, volume 2298, pages 34–46. Springer, 2001. [cited at p. 78]

[34] Ingemar J. Cox : A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993. [cited at p. 12]

[35] Navneet Dalal et Bill Triggs : Histograms of oriented gradients for human detection. *In International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005. [cited at p. 94]

[36] Patrice Dalle, Annelies Braffort et Christophe Collet : Accessibilité et langue des signes : modélisations, méthodes, application (regular (long) paper). *Conférence Internationale sur l'accessibilité et les systèmes de suppléance aux personnes en situations de handicaps (ASSISTH), Toulouse, 19/11/07-21/11/07*, pages 209–217, 2007. [cited at p. 2, 34]

[37] A. P. Dempster, N. M. Laird et D. B. Rubin : Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society : Series B*, 39:1–38, 1977. [cited at p. 12, 65]

[38] Shaolei Feng, R. Manmatha et Andrew Mccallum : Exploring the use of conditional random field models and hmms for historical handwritten document recognition. *In the Proceedings of the 2nd IEEE International Conference on Document Image Analysis for Libraries (DIAL)*, pages 30–37, 2006. [cited at p. 77]

[39] J.D. Ferguson : Variable duration models for speech. page 143 ?179, Princeton, NJ, 1980. [cited at p. 63, 67]

[40] Rogerio S. Feris, Volker Krueger et Roberto M. Cesar, Jr. : A wavelet subspace method for real-time face tracking. *Real-Time Imaging*, 10(6):339–350, décembre 2004. [cited at p. 12]

[41] H. Freeman : On the encoding of arbitrary geometric configurations. *Institute of Radio Engineers, trans. on Electronic Computers*, EC-10:260–268, 1961. [cited at p. 28]

[42] Keinosuke Fukunaga et Larry D. Hostetler : The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975. [cited at p. 132]

[43] Mary M. Galloway : Texture analysis using gray level run lengths. *Computer Graphics and Image Processing*, 4(2):172 – 179, 1975. [cited at p. 27]

[44] Aravind Ganapathiraju, Jonathan Hamaker et Joseph Picone : Hybrid svm/hmm architectures for speech recognition. *In INTERSPEECH*, pages 504–507. ISCA, 2000. [cited at p. 78]

[45] Frédérick Gianni : *Suivi de parties du corps pour l'interprétation de gestes de communication à partir de séquence monoculaire.* Thèse de doctorat, 2007. [cited at p. 24]

[46] Frédérick Gianni, Christophe Collet et François Lefebvre : Modèles et méthodes de traitement d'images pour l'analyse de la langue des signes. *TAL*, 48(3):175–200, 2007. [cited at p. 25, 26, 27, 57]

[47] S. Gidel, Aubiere, C. Blanc, T. Chateau et P. Checchin : Nonparametric data association for particle filter based multi-object tracking : application to multi-pedestrian tracking. *Intelligent Vehicles Symposium, IEEE*, pages 73 – 78, June 2008. [cited at p. 12]

[48] Michel Gilloux, Bernard Lemarié et Manuel Leroux : A hybrid radial basis function network/hidden markov model handwritten word recognition system. *In ICDAR*, pages 394–397. IEEE Computer Society, 1995. [cited at p. 78]

[49] N. J. Gordon, D. J. Salmond et A. F. M. Smith : Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, Août 2002. [cited at p. 6, 12, 15]

[50] Philippe Gérard et André Gagalowicz : Three dimensional model-based tracking using texture learning and matching. *Pattern Recognition Letters*, 21(13-14):1095–1103, 2000. [cited at p. 12]

[51] W. E. L. Grimson, C. Stauffer, R. Romano et L. Lee : Using adaptive tracking to classify and monitor activities in a site. CVPR '98, pages 22–, Washington, DC, USA, 1998. IEEE Computer Society. [cited at p. 29]

[52] K. Grobel et M. Assam : Isolated sign language recognition using hidden markov models. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, page 162Ű167, 1997. [cited at p. 68]

[53] Asela Gunawardana, Milind Mahajan, Alex Acero et John C. Platt : Hidden conditional random fields for phone classification. *In INTERSPEECH*, pages 1117–1120. ISCA, 2005. [cited at p. 74, 78]

[54] Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, Ben Hamner et Hugo Jair Escalante : Chalearn gesture challenge : Design and first results. *In CVPR Workshops*, pages 1–6. IEEE, 2012. [cited at p. 2, 3, 7, 98]

[55] Robert M. Haralick, K. Shanmugam et Its'Hak Dinstein : Textural Features for Image Classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3(6):610–621, novembre 1973. [cited at p. 27]

[56] Chris Harris et Mike Stephens : A combined corner and edge detector. *In In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988. [cited at p. 28]

[57] David Hébert : *Champs aléatoires conditionnels pour l'extraction de structures dans les images de documents*. Thèse de doctorat, Université de Rouen, 2013. [cited at p. 63, 77]

[58] A. J. H. Hii, Christopher E. Hann, J. Geoffrey Chase et Eli E. W. Van Houten : Fast normalized cross correlation for motion tracking using basis functions. *Computer Methods and Programs in Biomedicine*, 82(2):144–156, 2006. [cited at p. 12]

[59] Ming-Kuei Hu : Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, February 1962. [cited at p. 28, 29]

[60] Michael Isard et Andrew Blake : Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998. [cited at p. 12, 20, 23, 24, 26, 35]

[61] Paul Jaccard : *Bulletin de la Société Vaudoise des Sciences Naturelles*, volume 37. 1901. [cited at p. 31]

[62] Eric Jackson : An hmm-based approach for gesture recognition using edge features. CVPR 2012 Workshop on Gesture Recognition, http://gesture.chalearn.org/dissemination/cvpr2012, June 2012. [cited at p. 68, 84, 96, 105]

[63] Ramesh Jain et H. H. Nagel : On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(2):206–214, feb 1979. [cited at p. 29]

[64] A. Jepson et M.J. Black : Mixture models for optical flow computation. *In Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pages 760–761, 1993. [cited at p. 12]

[65] Finn Tore Johansen : A comparison of hybrid hmm architectures using global discriminative training. *In ICSLP*. ISCA, 1996. [cited at p. 78]

[66] Kalman, Rudolph et Emil : A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960. [cited at p. 12, 14, 129]

[67] Shunsuke Kamijo et Masao Sakauchi : Illumination invariant and occlusion robust vehicle tracking by spatio-temporal mrf model. *In In Proc. 9th World Congress on ITS*, 2002. [cited at p. 12]

[68] Michael Kass, Andrew Witkin et Demetri Terzopoulos : Snakes : Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4): 321–331, 1988. [cited at p. 28]

[69] Hae-Kwang Kim et Jong-Deuk Kim : Region-based shape descriptor invariant to rotation, scale and translation. *Signal Processing : Image Communication*, 16(1–2):87–93, 2000. [cited at p. 29]

[70] Genshiro Kitagawa : Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996. [cited at p. 12, 15]

[71] John Klein : *Suivi des objets dans des séquences d'images par fusion de sources : application au suivi de véhicules dans des scènes routières.* Thèse de doctorat, 2008. [cited at p. 12, 15, 16, 18, 19, 20, 24, 27, 28, 29, 31, 43]

[72] S. Knerr et E. Augustin : A neural network-hidden markov model hybrid for cursive word recognition. *ICPR*, 2:1518–1520, 1998. [cited at p. 78]

[73] Jakub Konencny et Michal Hagara : One-shot learning gesture recognition using hog/hof features. ICPR 2012 Workshop on Gesture Recognition, http://gesture.chalearn.org/dissemination/icpr2012, Novembre 2012. [cited at p. 68, 69, 84, 96, 105]

[74] John D. Lafferty, Andrew McCallum et Fernando C. N. Pereira : Conditional random fields : Probabilistic models for segmenting and labeling sequence data. *In Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. [cited at p. 63, 70]

[75] Ivan Laptev, Serge J. Belongie, Patrick Perez et Josh Wills : Periodic motion detection and segmentation via approximate sequence alignment. *In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1 - Volume 01*, ICCV '05, pages 816–823, Washington, DC, USA, 2005. IEEE Computer Society. [cited at p. 29]

[76] C. Lemaître : Utilisation de masques binaries dans une rétine de recon- naissance des formes. Mémoire de D.E.A., 2005. [cited at p. 29]

[77] Boris Lenseigne, Frédérick Gianni et Patrice Dalle : A new gesture representation for sign language analysis. *LREC 2004 - Workshop on the Representation and Processing of Sign Language : From SignWriting to Image Processing , Lisbonne, Portugal, 25/05/2004-30/05/2004*, pages 85–90, mai 2004. [cited at p. 57]

[78] VI Levenshtein : Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, 1966. [cited at p. 99]

[79] Peihua Li, Tianwen Zhang et Bo Ma : Unscented kalman filter for visual curve tracking. *Image and Vision Computing*, 22(2):157 – 164, 2004. Statistical Methods in Video Processing. [cited at p. 131]

[80] Tsong-wuu Lin : A comparative study of zernike moments. *Intelligence*, 1:4–7, 2003. [cited at p. 42]

[81] Jose Luis Lisani et Jean-Michel Morel : Detection of major changes in satellite images. *In ICIP (1)*, pages 941–944, 2003. [cited at p. 29]

[82] D. C. Liu et J. Nocedal : On the limited memory bfgs method for large scale optimization. *Math. Program.*, 45(3):503–528, dec 1989. [cited at p. 72]

[83] Jun S. Liu et Rong Chen : Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998. [cited at p. 18]

[84] Wei-Lwun Lu, Kenji Okuma et James J. Little : Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image Vision Comput.*, 27(1-2):189–205, jan 2009. [cited at p. 24]

[85] Bruce D. Lucas et Takeo Kanade : An iterative image registration technique with an application to stereo vision. *In Int Joint Conf Artif Intel*, volume 2 de *IJCAI'81*, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc. [cited at p. 29, 34, 45]

[86] John MacCormick et Michael Isard : Partitioned sampling, articulated objects, and interface-quality hand tracking. volume 1843, pages 3–19. Springer, 2000. [cited at p. 24]

[87] Sanparith Marukatat, Thierry Artières, Patrick Gallinari et Bernadette Dorizzi : Sentence recognition through hybrid neuro-markovian modeling. *In ICDAR*, pages 731–. IEEE Computer Society, 2001. [cited at p. 78]

[88] Ofer Matan, Christopher J. C. Burges, Yann Le Cun et John S. Denker : Multi-digit recognition using a space displacement neural network. *In Advances in Neural Information Processing Systems 4*, pages 488–495. Morgan Kaufmann, San Francisco, CA, 1992. [cited at p. 78]

[89] Rajiv Mehrotra et James E. Gary : Similar-shape retrieval in shape data management. *IEEE Computer*, 28(9):57–62, 1995. [cited at p. 28]

[90] Magdi A. Mohamed et Paul D. Gader : Generalized hidden markov models. ii. application to handwritten word recognition. *IEEE T. Fuzzy Systems*, 8(1):82–94, 2000. [cited at p. 68]

[91] Louis-Philippe Morency, Ariadna Quattoni et Trevor Darrell : Latent-dynamic discriminative models for continuous gesture recognition. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007. [cited at p. 63, 75, 76, 77]

[92] N. Morgan, H. Bourlard, S. Renls, M. Cohen et H. Franco : Hybrid neural network/hidden markov model systems for continuous speech recognition. *IJPRAI*, 7(4), 1993. [cited at p. 78]

[93] Marisa E. Morita, Robert Sabourin, Flávio Bortolozzi et Ching Y. Suen : Segmentation and recognition of handwritten dates : an hmm-mlp hybrid approach. *IJDAR*, 6(4):248–262, 2003. [cited at p. 78]

[94] C. Eddie Moxey, Stephen J. Sangwine et Todd A. Ell : Hypercomplex correlation techniques for vector images. *IEEE Transactions on Signal Processing*, 51(7):1941–1953, 2003. [cited at p. 27]

[95] Shibaji Mukherjee et Sushmita Mitra : Hidden Markov Models, grammars, and biology : a tutorial. *Journal of bioinformatics and computational biology*, 3(2):491–526, avril 2005. [cited at p. 68]

[96] Carol Neidle, Judy Kegl, Dawn MacLaughli, Benjamin Bahan et Robert G. Lee : *The Syntax of American Sign Language-Functional Categories and Hierarchical Structure*. MIT Press, 2000. [cited at p. 34]

[97] Carol Neidle, Stan Sclaroff et Vassilis Athitsos : Signstream : A tool for linguistic and computer vision research on visual-gestural language data. *Behavior Research Methods, Instruments, & Computers*, 33(3):311–320, 2001. [cited at p. 7, 34, 48, 92]

[98] H. T. Nguyen et A. W. M. Smeulders : Fast occluded object tracking by a robust appearance filter. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1099–1104, 2004. [cited at p. 12]

[99] L. T. Niles et H. F. Silverman : Combining hidden Markov models and neural network classifiers. *In Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 417–420, Albuquerque, 1990. [cited at p. 78]

[100] J. M. Odobez et P. Bouthemy : Robust multiresolution estimation of parametric motion models. *Jal of Vis. Comm. and Image Representation*, 1995. [cited at p. 29]

[101] Surendra Ong, Sylvie C. W.and Ranganath : Deciphering gestures with layered meanings and signer adaptation. *Automatic Face and Gesture Recognition, IEEE International Conference on*, 0:559, 2004. [cited at p. 68, 84]

[102] V. Pashaloudi et Margaritis K.G. : Feature extraction and sign recognition for greek sign language. *Proceedings of the 11th IEEE Mediterranean Conference on Control and Automation*, 2003. [cited at p. 68]

[103] Greg Pass, Ramin Zabih et Justin Miller : Comparing images using color coherence vectors. *In Proceedings of the Fourth ACM International Conference on Multimedia*, MULTIMEDIA '96, pages 65–73, New York, NY, USA, 1996. ACM. [cited at p. 27]

[104] B V Patel et B B Meshram : Content based video retrieval systems. *International Journal of UbiComp (IJU)*, 3(2), April 2012. [cited at p. 4]

[105] Patrick Perez, Jaco Vermaak et Andrew Blake : Data fusion for visual tracking with particles. *In Proceedings of the IEEE*, pages 495–513, 2004. [cited at p. 26]

[106] Michael K. Pitt et Neil Shephard : Filtering via Simulation : Auxiliary Particle Filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999. [cited at p. 24]

[107] J. M. S. Prewitt : Object enhancement and extraction. *Picture Processing and Psichopictorics, Academic Press*, 1970. [cited at p. 28]

[108] Ariadna Quattoni, Michael Collins et Trevor Darrell : Conditional random fields for object recognition. *In In NIPS*, pages 1097–1104. MIT Press, 2004. [cited at p. 63, 73, 74, 75]

[109] Lawrence R. Rabiner : A tutorial on hidden markov models and selected applications in speech recognition. *In Proceedings of the IEEE*, pages 257–286, 1989. [cited at p. 5, 6, 63, 68, 72]

[110] Lawrence R. Rabiner : Readings in speech recognition. chapitre A tutorial on hidden Markov models and selected applications in speech recognition, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. [cited at p. 65]

[111] Stjepan Rajko et Gang Qian : A hybrid hmm/dpa adaptive gesture recognition method. *In ISVC*, volume 3804, pages 227–234. Springer, 2005. [cited at p. 78]

[112] D. Reid : An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, dec 1979. [cited at p. 12]

[113] Ying Ren, Chin-Seng Chua et Yeong-Khing Ho : Statistical background modeling for non-stationary camera. *Pattern Recognition Letters*, 24(1 ?3):183 – 196, 2003. [cited at p. 29]

[114] Jérôme Revaud, Guillaume Lavoué et Atilla Baskurt : Une nouvelle mesure de distance entre descripteurs de moments de Zernike pour une similarité optimale et un angle de rotation entre les images. *In CORESA*, mars 2009. [cited at p. 28, 44]

[115] Gerhard Rigoll : Maximum mutual information neural networks for hybrid connectionist-hmm speech recognition systems. *IEEE Transactions on Speech and Audio Processing*, 2(1):175–184, 1994. [cited at p. 78]

[116] Branko Ristic, Sanjeev Arulampalam et Neil Gordon : *Beyond the Kalman Filter : Particle Filters for Tracking Applications*. Artech House, 2004. [cited at p. 15]

[117] Cyril Royère, Dominique Gruyer et Véronique Cherfaoui : Data association with believe theory. *Int. Conf. on Information Fusion*, 1, 2000. [cited at p. 12]

[118] Y. Rubner, C. Tomasi et L. J. Guibas : The earth mover's distance as a metriccfor image retrieval. *International Journal of Computer Vision*, 40, 2000. [cited at p. 44]

[119] Sunita Sarawagi et William W. Cohen : Semi-markov conditional random fields for information extraction. pages 1185–1192, 2004. [cited at p. 63, 72]

[120] M. Sarfraz, Yusuf A. Syed et M. Zeeshan : A system for sign language recognition using fuzzy object similarity tracking. *In IV '05 : Proceedings of the Ninth International Conference on Information Visualisation*, pages 233–238, Washington, DC, USA, 2005. IEEE Computer Society. [cited at p. 68]

[121] Kenneth M. Sayre : Machine recognition of handwritten words : A project report. *Pattern Recognition*, 5(3):213 – 228, 1973. [cited at p. 4, 62]

[122] I. K. Sethi et Ramesh Jain : Finding trajectories of feature points in a monocular image sequence. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(1):56–73, janvier 1987. [cited at p. 12]

[123] Caifeng Shan, Tieniu Tan et Yucheng Wei : Real-time hand tracking using a mean shift embedded particle filter. *PR*, 40(7):1958–1970, July 2007. [cited at p. 24, 25]

[124] Jianbo Shi et Jitendra Malik : Motion segmentation and tracking using normalized cuts. *In ICCV*, pages 1154–1160, 1998. [cited at p. 12]

[125] Leonid Sigal, Stan Sclaroff et Vassilis Athitsos : Skin color-based video segmentation under time-varying illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:862–877, 2003. [cited at p. 28, 40]

[126] Y. Soullard et T. Artières : Hybrid hmm and hcrf model for sequence classification. Bruges (Belgium), April 2011. [cited at p. 78]

[127] Bjorn Stenger, Arasanathan Thayananthan, Philip H. S. Torr et Roberto Cipolla : Model-based hand tracking using a hierarchical bayesian filter. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1372–1384, septembre 2006. [cited at p. 24]

[128] William Stokoe : Sign language structure : An online of the visual communication systems of the american deaf. 1960. *Journal of Deaf Studies and Deaf Education*, 10:3–37, 2005. [cited at p. 2, 34]

[129] Michael J. Swain et Dana H. Ballard : Color indexing. *Int. J. Comput. Vision*, 7(1):11–32, novembre 1991. [cited at p. 27, 133]

[130] Nobuhiko Tanibata, Nobutaka Shimada et Yoshiaki Shirai : Extraction of hand features for recognition of sign language words. *In In International Conference on Vision Interface*, pages 391–398, 2002. [cited at p. 68]

[131] Joe Tebelskis, Alex Waibel, Bojan Petek et Otto Schmidbauer : Continuous speech recognition by linked predictive neural networks. *In NIPS*, pages 199–205. Morgan Kaufmann, 1990. [cited at p. 78]

[132] S. Thomas, C. Chatelain, L. Heutte et T. Paquet : Combinaison architecture profonde/hmm pour l'extraction de sequences dans des documents manuscrits. *In CIFED, Bordeaux, France*, 2012. [cited at p. 78]

[133] Péter Torma et Csaba Szepesvári : Enhancing particle filters using local likelihood sampling. *In ECCV (1)*, volume 3021 de *Lecture Notes in Computer Science*, pages 16–27. Springer, 2004. [cited at p. 24]

[134] E. Trentin : A survey of hybrid ANN/HMM models for automatic speech recognition. *Neurocomputing*, (1-4):91–126, avril 2001. [cited at p. 78]

[135] M. Unser : Texture classification and segmentation using wavelet frames. *Image Processing, IEEE Transactions on*, 4(11):1549–1560, novembre 1995. [cited at p. 27]

[136] T. L. M. van Kasteren, G. Englebienne et B. J. A. Kröse : Activity recognition using semi-markov models on real world smart home datasets. *J. Ambient Intell. Smart Environ.*, 2(3):311–325, aug 2010. [cited at p. 66, 68, 69, 73, 77]

[137] Jim H. Velduis et G. Wayne Brodland : A deformable block-matching algorithm for tracking epithelial cells. *Image Vision Comput.*, 17(12):905–911, 1999. [cited at p. 12]

[138] René Vidal et Yi Ma : A unified algebraic approach to 2-d and 3-d motion segmentation. *In ECCV (1)*, volume 3021, pages 1–15. Springer, 2004. [cited at p. 29]

[139] Paul Viola et Michael J. Jones : Robust Real-Time face detection. *Int J Comput Vison*, 57(2):137–154, may 2004. [cited at p. 36, 41]

[140] A. Viterbi : Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, apr 1967. [cited at p. 62, 65]

[141] Christian Vogler et Dimitris Metaxas : A framework for recognizing the simultaneous aspects of american sign language. *Computer Vision and Image Understanding*, 81:358–384, 2001. [cited at p. 2, 34, 68, 84]

[142] Ulrich von Agris, Jörg Zieren, Ulrich Canzler, Britta Bauer et Karl-Friedrich Kraiss : Recent developments in visual sign language recognition. *Univers. Access Inf. Soc.*, 6(4):323–362, 2008. Institute of Man Machine Interaction, RWTH Aachen University. [cited at p. 2, 34, 68, 69, 84]

[143] Hanna M. Wallach : Conditional random fields : An introduction. Rapport technique, University of Pennsylvania CIS Technical Report MS-CIS-04-21, February 2004. [cited at p. 5, 6, 63]

[144] Eric A. Wan et Rudolph Van Der Merwe : The unscented kalman filter for nonlinear estimation. pages 153–158, 2000. [cited at p. 24]

[145] Sy Bor Wang, Ariadna Quattoni, Louis-Philippe Morency et David Demirdjian : Hidden conditional random fields for gesture recognition. *In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, pages 1521–1527, Washington, DC, USA, 2006. IEEE Computer Society. [cited at p. 75, 77]

[146] David Weiss : Hmm based one shot gesture recognition. CVPR 2012 Workshop on Gesture Recognition, http://gesture.chalearn.org/dissemination/cvpr2012, June 2012. [cited at p. 68, 84, 96, 105]

[147] Lloyd R. Welch : Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4), December 2003. [cited at p. 65]

[148] L. Wixson et M. Hansen : Detecting salient motion by accumulating directionally-consistent flow. *In Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 797–, Washington, DC, USA, 1999. IEEE Computer Society. [cited at p. 29]

[149] Di Wu, Fan Zhu et Ling Shao : One shot learning gesture recognition from rgbd images. pages 7–12. CVPR, IEEE, 2012. [cited at p. 96]

[150] Hee-Deok Yang et Seong-Whan Lee : Robust sign language recognition with hierarchical conditional random fields. *Pattern Recognition, International Conference on*, 0:2202–2205, 2010. [cited at p. 77]

[151] Hee-Deok Yang, Stan Sclaroff et Seong-Whan Lee : Sign language spotting with a threshold model based on conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1264–1277, 2009. [cited at p. 77]

[152] Shuying YANG, Weimin GE, Cheng ZHANG et Pilian HE : Detecting and tracking moving targets on omnidirectional vision. *Tianjin University and Springer-Verlag*, 15:013–018, 2009. [cited at p. 26]

[153] Yang YANG, I. SALEEMI et M. SHAH : Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1635–1648, 2013. [cited at p. 7, 68, 96]

[154] Angela YAO, Dominique UEBERSAX, Juergen GALL et Luc VAN GOOL : Tracking people in broadcast sports. *In DAGM-PR*, pages 151–161, Berlin, Heidelberg, 2010. Springer-Verlag. [cited at p. 24, 25]

[155] Tae-Woong YOO et Il-Seok OH : A fast algorithm for tracking human faces based on chromatic histograms. *Pattern Recogn. Lett.*, 20(10):967–978, octobre 1999. [cited at p. 27]

[156] Quan YUAN, Stan SCLAROFF et Vassilis ATHITSOS : Automatic 2d hand tracking in video sequences. *In Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05) - Volume 1 - Volume 01*, pages 250–256, Washington, DC, USA, 2005. IEEE Computer Society. [cited at p. 54]

[157] Benjamin D. ZARIT, Boaz J. SUPER et Francis K. H. QUEK : Comparison of five color models in skin pixel classification. *In In ICCV'99 Int'l Workshop on*, pages 58–63, 1999. [cited at p. 28, 40]

[158] George ZAVALIAGKOS, Steve AUSTIN, John MAKHOUL et Richard M. SCHWARTZ : A hybrid continuous speech recognition system using segmental neural nets with hidden markov models. *IJPRAI*, 7(4):949–963, 1993. [cited at p. 78]

[159] D. ZHANG et G. LU : Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004. [cited at p. 28]

[160] Steven W ZUCKER : Toward a model of texture. *Computer Graphics and Image Processing*, 5(2):190 – 202, 1976. [cited at p. 27]

Annexe A

Principe d'autres méthodes de suivi

Table des matières

A.1 Filtre de Kalman

A.1.1 Théorie

Les filtres de `Kalman` [66] définissent les fonctions de transition f et d'observation g d'une manière déterministe linéaire selon les équations suivantes :

$$Y_t = F_t Y_{t-1} + b_t \quad (A.1)$$

$$X_t = G_t Y_t + c_t \quad (A.2)$$

où F_t et G_t sont deux matrices et b_t et c_t sont deux vecteurs de bruits. Si les bruits sont gaussiens, le filtre de `Kalman` donne une réponse optimale au problème. Le but est d'estimer Y_t à chaque instant t selon deux procédures : une procédure de `prédiction` (équation A.3) et une procédure de `correction` (équation A.4).

$$\tilde{Y}_t = F_t \hat{Y}_{t-1} \quad (A.3)$$
$$\hat{Y}_t = \tilde{Y}_t + k_t(X_t - G_t \tilde{Y}_t) \quad (A.4)$$

où $\tilde{Y}$ est une estimation intermédiaire prédite par l'équation de transition A.3 et k_t est un scalaire appelé gain de Kalman, il permet de favoriser la prédiction ou l'observation selon sa valeur. Le coefficient $X_t - G_t F_t \hat{Y}_{t-1}$ est appelé innovation.

Les terme b_t de l'équation de prédiction et c_t de l'équation de correction peuvent s'écrire $b_t = b'_t + b''_t$ et $c_t = c'_t + c''_t$. b'_t et c'_t sont deux vecteurs et b''_t et $c"_t$ sont deux vecteurs de bruits blancs gaussiens centrés. F_t, G_t, b'_t et c'_t sont des données du système connues *a priori*. Les matrices de covariances de b''_t et c''_t respectivement notées Q_t et R_t sont supposées connues également. Les bruits sont par hypothèse indépendants. Y_0 est également connu. Le système peut s'écrire sous forme probabiliste :

$$Y_0 \sim \mathcal{N}(Y_0, \bar{Y}_0, \Sigma_0) \quad (A.5)$$
$$Y_t|Y_{t-1} \sim \mathcal{N}(Y_t, F_t Y_t + b'_t, Q_t) \quad (A.6)$$
$$X_t|Y_t \sim \mathcal{N}(X_t, H_t Y_t + c'_t, R_t) \quad (A.7)$$

avec $\forall t$, $\bar{Y}_t$ est l'espérance de Y_t et Σ_t est la covariance de Y_t. L'espérance et la covariance suffisent pour caractériser entièrement la densité de filtrage, ce sont alors les paramètres à estimer avec le filtre de Kalman. Les deux étapes principales du filtre de Kalman définies par les équations de prédiction et de correction sont ainsi formulées comme suit où l'indice $t|t'$ désigne que la variable est calculée au temps t sachant les t' premiers valeurs :

Étape de prédiction

$$\bar{Y}_{t|t-1} = F_t \bar{Y}_{t-1|t-1} + b'_t \quad (A.8)$$
$$\Sigma_{t/t-1} = F_t \Sigma_{t-1} F_t^T + Q_t \quad (A.9)$$

Étape de correction

$$K_t = \Sigma_{t|t-1} G_t^T (G_t \Sigma_{t|t-1} G_t^T + R_T) \quad (A.10)$$
$$\bar{Y}_{t|t} = \bar{Y}_{t|t-1} + K_t[Y_t - (G_t \bar{X}_{t|t-1} + c_t)] \quad (A.11)$$
$$\Sigma_{t|t} = (Id - K_t G_t)\Sigma t|t-1 \quad (A.12)$$

Lorsqu'on présente le filtre de Kalman en utilisant la théorie de l'estimation, la matrice $\Sigma_{t|t}$ est souvent appelée matrice de covariance de l'erreur. La trace de cette matrice est égale à l'erreur au sens des moindres carrés entre l'état et son estimée.

Ce que nous venons de présenter est la variante simple des filtres de Kalman caractérisée par un calcul linéaire. Cette variante estime la loi conditionnelle d'un processus d'état sachant un processus observation d'une manière exacte et rapide lorsque les dynamiques de l'état et de l'observation sont linéaires et gaussiennes. Dans le cas général, lorsque la contrainte de linéarité est difficile à satisfaire, deux autres variantes des filtres de Kalman sont proposées avec une modélisation non-linéaire plus flexible.

A.1.2 Variantes

Les `filtres de Kalman étendus` reposent sur une modélisation non-linéaire des relations de transition (prédiction) et d'observation (correction). Le développement de Taylor est utilisé pour approximer les fonctions f_t et g_t par des modèles non-linéaires. F_t et G_t sont définies par des matrices Jacobiennes :

$$F_t \approx \left.\frac{df}{dY_t}\right|_{Y_t=\hat{Y}_{t-1}} \tag{A.13}$$

$$G_t \approx \left.\frac{dg}{dY_t}\right|_{Y_t=\hat{Y}_{t-1}} \tag{A.14}$$

Néanmoins, le calcul de ces matrices Jacobiennes est coûteux et l'approximation de Taylor n'est pas toujours satisfaisante.

Le `filtre de Kalman sans parfum` (unscented Kalman filter) tente de surmonter ces défauts en modélisant la non-linéarité des relations de transition et d'observation à travers une approximation de la distribution de filtrage à l'aide de l'échantillonnage déterministe sigma-points. Le formalisme d'échantillonnage sigma-points calcule $2n+1$ échantillons $y_{t-1}^{(i)}$ et un poids associé $w_{t-1}^{(i)}$ tel que n est la dimension du vecteur Y_t :

$$\begin{cases} y_{t-1}^{(0)} = \bar{Y}_{t-1} & w_0 = \frac{k}{n+k},\ i=0 \\ y_{t-1}^{(i)} = \bar{Y}_{t-1} + (\sqrt{(n+k)+\Sigma_{Y_{t-1}}}) & w_i = \frac{1}{2(n+k)},\ 1 \leq i \leq n \\ y_{t-1}^{(i)} = \bar{Y}_{t-1} - (\sqrt{(n+k)+\Sigma_{Y_{t-1}}}) & w_i = \frac{1}{2(n+k)},\ (n+1) \leq i \leq 2n \end{cases} \tag{A.15}$$

L'ensemble d'échantillons à l'instant t, y_t, dépend de l'ensemble des échantillons à l'instant $t-1$, y_{t-1}, d'une manière déterministe à travers une fonction non-linéaire f : $y_t^{(i)} = f(y_{t-1}^{(i)})$. En supposant que $Y_t|Y_{t-1}$ suit une loi Gaussienne, il suffit de calculer la moyenne $\bar{Y}_t$ et la covariance Σ_t de Y_t :

$$\bar{Y}_t = \sum_{i=0}^{2n} w_t^{(i)} y_t^{(i)} \tag{A.16}$$

$$\Sigma_t = \sum_{i=0}^{2n} w_t^{(i)} \left[(y_t^{(i)} - \bar{Y}_t)(y_t^{(i)} - \bar{Y}_t)^T \right] + Q_t \tag{A.17}$$

Cette procédure de transformation sans parfum est appliquée également à l'équation d'observation (2.4).

Li et al [79] ont appliqué le filtre de Kalman sans parfum au suivi des objets par contour. Leurs expériences montrent que les performances de cette variante sont meilleures que les performances des filtres de Kalman. Cependant, ils concluent que les filtres particulaires donnent des résultats encore meilleurs.

Le filtre particulaire se base d'une manière analogue au filtre de Kalman sans parfum sur le principe d'échantillonnage sauf que la génération des échantillons à l'instant t dépend des échantillons calculés à l'instant $t-1$ d'une manière implicite. L'échantillonnage des $Y_t^{(i)}$ suit une densité de probabilité dont la moyenne est calculée en fonction des $Y_{t-1}^{(i)}$. Dans le cas des filtres de Kalman sans parfum, les échantillons $Y_t^{(i)}$ s'expriment explicitement en fonction des échantillons $Y_{t-1}^{(i)}$ à travers une fonction f.

A.2 Camshift : *Continously Adaptative Mean Shift*

A.2.1 Principe du Camshift

La méthode du Camshift [20] se base sur le même principe que la méthode de Mean-Shift, présentée dans la sous-section A.2.2, avec adaptation de la fenêtre de recherche. La méthode du Camshift peut être utilisée pour la segmentation d'un objet dans une image fixe comme elle peut être utilisée pour le suivi d'un objet sur une séquence d'images. Dans le premier cas, l'algorithme du Mean-Shift est lancé à chaque itération en faisant augmenter la taille de la fenêtre de recherche d'une itération à l'autre. L'algorithme 6 représente l'algorithme du Camshift sur une image fixe.

Initialisation, $t = 0$:
Initialiser max_0
tant que $max_{t+1} - max_t > \epsilon_1$ **faire**
  $l = 0$; **tant que** $max_t^{(l+1)} - max_t^{(l)} > \epsilon_2$ **faire**
    Calculer pour tout $x_i \in V(max_t^{(l)})$ le poids $K(x_i - max_t^{(l)}) = e^{c||x_i - max_t^{(l)}||^2}$
    réestimer le maxima $max_t^{(l+1)} = \frac{\sum_{x_i \in V(max_t^{(l)})} K(x_i - max_t^{(l)}) x_i}{\sum_{x_i \in V(max_t^{(l)})} K(x_i - max_t^{(l)})}$
  fin tq
  $V(max_{t+1}) = V(max_t^{(l)}) \cup V'(max_t^{(l)})$
fin tq
Fin

Algorithm 6: Camshift sur une image fixe

$V'(max_t)$ est un sous-ensemble de l'ensemble complémentaire de $V(max_t)$. Dans le cas de la segmentation dans une images, $V'(max_t)$ représente un agrandissement de la fenêtre de recherche.

Dans le cas du suivi, le Camshift peut être lancé pour chaque image en prenant comme position initiale du maxima la position de l'objet suivi dans l'image précédente.

A.2.2 Principe du Mean-Shift

Le Mean-Shift [42] est une technique non-paramétrique d'analyse d'un espace de caractéristiques. Elle permet de localiser le maxima max (mode) d'une fonction de probabilité à partir d'un ensemble d'échantillons. Son principe se base sur une estimation itérative du maxima en utilisant une fonction noyau K qui est en général gaussienne (algorithme 7).

Initialisation, $t = 0$:
Initialiser max_0
tant que $max_{t+1} - max_t > \epsilon$ **faire**
  Calculer pour tout $x_i \in V(max_t)$ le poids $K(x_i - max_t) = e^{c||x_i - max_t||^2}$
  réestimer le maxima $max_{t+1} = \frac{\sum_{x_i \in V(max_t)} K(x_i - max_t) x_i}{\sum_{x_i \in V(max_t)} K(x_i - max_t)}$
fin tq
Fin

Algorithm 7: Mean-Shift

$V(max_t)$ est un voisinage de max_t où pour tout $x_i \in V(max_t), K(x_i - max_t) \neq 0$. Ce voisinage se déplace avec le déplacement du maxima d'une itération à l'autre. Parmi les applications de la méthode Mean-Shift, nous pouvons citer la segmentation, la classification et le suivi. Dans le cas du suivi, on peut choisir par exemple la couleur de l'objet comme caractéristique de l'objet. Pour chaque trame, pour chaque pixel de l'image, un poids est calculé en fonction de l'histogramme couleur de l'objet. Dans le cas de la rétro-projection [129] par exemple, ce poids représente le nombre de pixels ayant la couleur de ce pixel. Ainsi, l'image rétro-projetée représente des valeurs élevées au niveau des régions ayant une couleur proche de la couleur de l'objet suivi. Cette image rétro-projetée présente l'entrée de l'algorithme Mean-shift. Le Mean-Shift cherche ensuite le peak des poids à proximité de l'ancienne position de l'objet dans l'image. L'algorithme 7 est lancé pour chaque trame où max_0 prend la position de l'objet dans la trame précédente. Le voisinage $V(max_t)$ est représenté par une *fenêtre de recherche de taille fixe* qui est déplacée sur l'image en recentrant son centre selon le principe de la méthode Mean-Shift.