N° d'ordre: 3834

THÈSE

Soutenue

devant l'Université de Rennes 1

pour obtenir

le grade de : ***Docteur de l'Université de Rennes 1***

Mention : Traitement du Signal et Télécommunications

par

Alexandre HERVIEU

Équipe d'accueil : Vista (INRIA - Rennes Bretagne Atlantique, IRISA)
École Doctorale : Mathématiques, Informatique, Signal, Électronique et Télécommunications
Composante universitaire : SPM

Titre de la thèse :

Analyse de trajectoires vidéos à l'aide de modélisations markoviennes pour l'interprétation de contenus

Soutenue le 5 mars 2009 devant la commission d'examen

M. :	Patrick	PÉREZ	Président
MM. :	Riccardo	LEONARDI	Rapporteurs
	Philippe	JOLY	
MM. :	Jean-Marc	ODOBEZ	Examinateurs
	Patrick	BOUTHEMY	
	Jean-Pierre	LE CADRE	

"Dans les époques révolutionnaires, ceux qui s'attribuent, avec un si étrange orgueil, le facile mérite d'avoir développé chez leurs contemporains l'essor des passions anarchiques, ne s'aperçoivent pas que leur déplorable triomphe apparent n'est dû surtout qu'à une disposition spontanée, déterminée par l'ensemble de la situation sociale correspondante."

Auguste Comte - ***Cours de philosophie positive***

Remerciements

Je tiens tout d'abord à remercier Patrick Pérez pour avoir accepté de présider ce jury de thèse. Je remercie Riccardo Leonardi et Philippe Joly pour avoir accepté de relire la thèse et de donner leurs points de vue sur le travail présenté. Je suis également reconnaissant envers Jean-Marc Odobez d'avoir accepté d'être examinateur de mon travail.

Je désire tout particulièrement remercier Patrick Bouthemy et Jean-Pierre Le Cadre pour leur encadrement complémentaire et sans failles, chacun ayant toujours été présent à sa manière. Au delà du seul aspect scientifique, je salue leurs qualités humaines qui m'ont permis de ne jamais perdre la volonté et la motivation nécessaire à une telle entreprise.

Je désire également remercier l'ensemble des personnes rencontrées pendant la thèse et ayant permis de mener à bien ce projet, notamment les collègues de l'équipe VISTA. Travailler dans un tel cadre, tant au niveau scientifique que relationnel, est une chance. En essayant de n'oublier personne, tout d'abord les anciens : Nico, sans qui je n'aurai jamais fait cette thèse, et Aurélie qui m'ont tout deux énormément aidé et soutenu, merci pour tout. Gwen, Vincent, Anne, Thomas B., Thomas V., Elise, Claire, Imran, Amaury, André. Ceux qui sont toujours là également, Thierry, Patrick H., Charles, Ivan, Anatole, Christophe, Guillaume, Emilie, Cécile. Un merci très spécial à Adrien, je n'oublierai pas l'écoute dont tu as su faire preuve ainsi que l'aide que tu as pu m'apporter, scientifique ou non. J'aurai aimer rester plus longtemps ton co-bureau, malheureusement, toutes les bonnes choses ont une fin... Je tiens également à remercier spécialement quelques TEMICS et autres, Gaël, le co-bureau des débuts et le support technique si important, Hervé, Denis, Bouclinette, Joaquim, Thomas et également Kévin. Un grand merci spécial à Huguette pour la préparation de la thèse, et surtout pour avoir toujours été là quand j'en avais besoin.

Les amis de Normandie ont également compté pendant ces années. Bien que je les ai peu vus, j'ai souvent pensé à vous. J'en oublie forcément, mais merci à Fonf, Julien T., Kevin, Ronan, Tonio, Dams notamment, ainsi que les ébroïciens Mathieu G. et ses parents, Simon, Charlotte, Émilie, Benjamin, Mantes, Pierre... et tellement d'autres que j'ai un peu perdu de vue. Merci à vous d'avoir été là pendant ces belles années d'étudiant.

Un merci idoinement dédié à Simon, ainsi qu'aux deux Romains pour les tournées mondiales, pour le chouff et tout le reste (ça ferait beaucoup à détailler, de plus belle la vie au golf en passant par les matchs de l'OM). Une pensée également pour l'ensemble des creuvards, ainsi qu'à la nouvelle génération qui arrive, ils se reconnaîtront et savent en quoi ils ont participé à cette période et à ce travail.

Merci à ma famille, papa, maman, Christelle, Sébastien, Arnaud et Manon, vous m'avez accompagné lors de ce travail et je ne vous remercierai jamais assez. Je tiens également à remercier Claude et Claudine, et Roger qui ont toujours été un soutien, et à avoir une pensée toute particulière pour Simone.

Enfin, je voudrais remercier ma pitite nanounette (sans oublier Romu, Minouche, Timounou, Marco, Bouba et Kiki) qui m'a accompagnée, soutenue, supportée quotidiennement, qui jamais ne m'a laissé tombée malgré les nombreuses difficultés. Je ne sais pas comment j'aurai pu mener à bien ces travaux sans ton soutien. J'espère vivre beaucoup d'autres défis à tes cotés.

Table des matières

Introduction générale

"Je comprends comment. Je ne comprends pas pourquoi."

George Orwell - ***1984***

La quantité d'images et de vidéos numériques produite est en perpétuelle croissance, les utilisateurs ne se limitant plus aux seuls professionnels de l'audiovisuel. En effet, chacun peut aujourd'hui, par le biais des nouveaux médias, produire ses propres flux de vidéos numériques. Ainsi, des besoins considérables apparaissent dans de nombreux domaines devant permettre l'exploitation et la consultation automatiques de ces quantités importantes de vidéos. En particulier, les informations recherchées concernent souvent la détection d'événements particuliers ainsi que la compréhension des activités observées. Des illustrations nombreuses peuvent en être trouvées dans les domaines de la vidéo-surveillance ou des vidéos de sport notamment.

Contrairement aux images fixes, les vidéos offrent un contenu par définition dynamique, devant fournir une information pertinente pour la compréhension d'activités. Afin d'illustrer l'importance du contenu dynamique pour l'interprétation de vidéos, Johansonn [Johansson 73] a montré que l'être humain peut reconnaître des actions et des activités à partir de données purement dynamiques. L'analyse des mouvements est donc une partie importante des recherches en vision par ordinateur, et regroupe de nombreuses problématiques. Mitiche et al. [Mitiche 96] ont ainsi tenté de dresser une liste des différents domaine de recherche formant l'analyse de mouvements dans des vidéos. Les principales problématiques à considérer sont :

- la détection du mouvement,
- la segmentation du mouvement,
- la mesure du mouvement sous forme paramétrique,

- la mesure de mouvement sous forme dense,
- le suivi de primitives ou de régions,
- la reconstruction 3D,
- l'interprétation du mouvement (classification),
- la reconnaissance d'activités et d'actions.

Parmi les huit problématiques définies ci-dessus, les cinq premières peuvent être utilisées pour l'extraction de trajectoires d'objets mobiles dans des vidéos. Nous nous proposons d'utiliser de telles trajectoires afin de concevoir des méthodes pour les deux dernières problématiques citées : l'interprétation du mouvement et la reconnaissance d'activités et d'actions. Notre objectif est d'élaborer, pour chacune des tâches considérées, une représentation adaptée, dans le cadre vidéo, des trajectoires observées. Nous développons également des modélisations probabilistes devant permettre de prendre en compte les propriétés des trajectoires et qui impliquent une procédure d'apprentissage. Leur intérêt est de pouvoir utiliser des outils statistiques à des fins de classification, de clustering (*i.e.*, de reconnaissance non-supervisée) et de détection d'évènements inattendus dans des plans vidéos à l'aide de trajectoires isolées. Puis, des tâches de reconnaissance de gestes et d'actions ainsi que de segmentation temporelle de vidéos seront envisagées, en utilisant alors les interactions entre objets mobiles observés.

Contributions

La première contribution de ce document consistera à établir un état de l'art des méthodes existantes pour l'utilisation, à des fins de reconnaissance de contenus, de séries temporelles et plus particulièrement de trajectoires issues de séquences vidéos. Cette synthèse permettra en effet de définir le cadre de travail choisi pour nos recherches.

Nos travaux comprennent tout d'abord la modélisation de trajectoires isolées de façon probabiliste. La représentation construite prend compte d'une classe d'invariances pertinentes, et exhibe des informations de dynamique et de forme des trajectoires. Nous développerons des modèles de Markov cachés originaux, reposant sur des quantifications des observations. Ces modélisations permettront de faire face à des ensembles de données qui peuvent être réduits. De plus, des procédures automatiques de sélection des paramètres impliqués seront proposées, tant au niveau de l'extraction des primitives que des modélisations probabilistes considérées. Nous exploitons ces différents éléments pour comparer des trajectoires isolées issues de plan extraits de vidéos de sport. Des tâches de classification, de clustering de plans vidéos, ainsi que de détection d'évènements inattendus seront aussi abordées.

Ensuite, notre volonté a été de définir des méthodes statistiques pour la compréhension de phénomènes décrits par plusieurs trajectoires observées simultanément. Ainsi, nous proposerons des représentations permettant, pour différentes probléma-

tiques telles que la reconnaissance d'actions ou de phases de vidéos de sport, de décrire et prendre en compte les interactions entre les trajectoires vidéos observées. Nous privilégierons des modélisations markoviennes hiérarchiques et parallèles, permettant d'inclure l'hypothèse de causalité temporelle des phénomènes observés pour la recherche automatique de contenus et pour la segmentation temporelles de vidéos. Nous mènerons une validation expérimentale conséquente pour valider les propriétés des méthodes définies dans différents contextes : classification et clustering de gestes et d'actions à l'aide de trajectoires 3D de parties du corps humain, segmentation et reconnaissance de phases de jeu de vidéos de squash et de handball à l'aide de trajectoires.

Description du document

Ce document est organisé en quatre parties selon le plan suivant.

Partie I : État de l'art et contexte

La première partie définit le cadre de nos travaux. Ainsi, elle se décompose en trois chapitres. Cette thèse se propose d'utiliser des modélisations markoviennes pour des tâches de reconnaissance supervisée et non-supervisée de contenus dans des vidéos. Ainsi, le chapitre 1 est dédié aux modélisations markoviennes, alors que le chapitre 2 est consacré aux méthodes de classification non supervisée de données. Le chapitre 3 pose le cadre considéré dans cette thèse à partir d'un état de l'art des travaux existants sur l'analyse des séries temporelles et plus particulièrement des trajectoires issues de vidéos.

Partie II : Reconnaissance d'évènements vidéos à l'aide de trajectoires

La seconde partie est dédiée à l'analyse de trajectoires extraites de plans vidéos à l'aide d'outils statistiques. Le premier chapitre de cette partie décrit la représentation de trajectoires développée. Cette représentation permet de mettre en valeur les propriétés de dynamique et de forme des trajectoires vidéos. De plus, la caractérisation de trajectoires introduite, respectant un ensemble d'invariances pertinentes, est adaptée au cadre vidéo considéré dans cette thèse et, notamment, aux trajectoires issues de vidéos acquises par des caméras mobiles. Dans le chapitre 4, nous proposons une modélisation probabiliste, inspirée des modèles de Markov cachés existants, et permettant de faire face à un ensemble limité de données pouvant apparaître lors du traitement de trajectoires vidéos. Ainsi, une distance entre trajectoires est construite qui sera exploitée pour des tâches de reconnaissance supervisée et non-supervisée. Des méthodes de sélection des paramètres impliqués sont également développées. Le

chapitre 5 contient les expérimentations réalisées pour la comparaison de trajectoires isolées, à partir des méthodes conçues dans cette partie. Des tâches de reconnaissance supervisée (classification) et non-supervisée (clustering) de contenus dans des vidéos de sport ont été effectuées, une extension à la reconnaissance de formes ainsi que la détection d'évènements inattendus.

Partie III : Reconnaissance de contenu vidéo par l'analyse des interactions entre trajectoires

La troisième partie de ce document traitera des méthodes mises en oeuvre pour l'analyse des interactions entre trajectoires issues de vidéos. Le chapitre 6 présente ainsi les aspects théoriques des représentations choisies et des modélisations markoviennes proposées pour la reconnaissance de contenus vidéos à l'aide de plusieurs trajectoires. Des tâches d'interprétation de vidéos de squash et de handball (*i.e.*, de reconnaissance et de segmentation simultanée de phases de sport) seront considérées. Le chapitre 7 portera sur l'application des méthodes développées dans le chapitre 6 dans le contexte vidéo. Les trajectoires de joueurs de handball et de squash sont données dans le plan du terrain de sport.

Une conclusion générale, permettant de faire une synthèse de l'étude réalisée et des perspectives liées à ces travaux sont données.

Enfin, des travaux portant sur la classification et le clustering de gestes et d'actions à l'aide de trajectoires 3D de parties du corps humain sont présentés en annexes. En effet, ces travaux nécessitent des validations supplémentaires. Les trajectoires 3D utilisées pour ces tâches de reconnaissance de gestes sont issues de reconstruction 3D à l'aide de plusieurs caméras, les trajectoires correspondant aux actions seront extraites à l'aide de procédés de capture de mouvement.

Première partie

État de l'art et contexte

Introduction

Depuis de nombreuses années, l'élaboration de techniques pour le suivi d'entités mobiles dans des vidéos est au coeur de nombreux travaux. L'objectif de cette thèse est d'utiliser les données résultant de ces procédures de suivi, et particulièrement les trajectoires, pour des tâches de reconnaissance de contenu dans des vidéos. Ainsi, cette première partie est consacrée à la mise en place de notre cadre de travail et des outils mathématiques principaux mis en place dans la suite de ce manuscrit. Cette première partie est structurée comme suit :

- le **chapitre 1** présente un état de l'art des principales modélisations markoviennes existantes. Ces méthodes permettent, par leur propriétés markoviennes, de prendre en compte explicitement les causalités temporelles inhérentes aux trajectoires. Ces modèles seront exploités dans l'ensemble de nos travaux ;

- Un court état de l'art des méthodes de clustering, tâche de reconnaissance considérée dans cette thèse, est également exposé en **chapitre 2** ;

- dans le **chapitre 3**, nous proposons, afin de mettre en place le cadre de travail choisi, un état de l'art des méthodes existantes de traitement des séries temporelles, et plus particulièrement des trajectoires observées dans des vidéos, dédiées à des tâches de reconnaissance.

Dans toute la suite de ce document, la reconnaissance supervisée pourra également aussi être appelée classification, la reconnaissance non-supervisée pouvant elle être également appelée clustering.

Chapitre 1

Modèles de Markov cachés

"Or, on avait remarqué que l'équation shannonienne de l'information était comme le reflet, le négatif, de celle de l'entropie dans le sens où l'entropie croît de manière inverse à l'information. [...] L'information est donc un concept qui établit le lien avec la physique tout en étant le concept fondamental inconnu de la physique. [...] L'information est un concept problématique, non un concept solution. C'est un concept indispensable, mais ce n'est pas encore un concept élucidé et élucidant. [...] les aspects émergés de la théorie de l'information, l'aspect communicationnel et l'aspect statistique, sont comme la mince surface d'un immense iceberg."

Edgar Morin - ***Introduction à la pensée complexe***

1.1 Introduction

Les vidéos étant des séquences d'images, le traitement de données séquentielles, ou séries temporelles, est d'une grande importance en vision par ordinateur. Deux approches "classiques" considérant les données séquentielles peuvent être envisagées, les modélisations externes (*i.e.*, l'analyse fréquentielle, les ondelettes ...) et les modélisations internes. Les modélisations internes de séries temporelles utilisent soit des modèles de régression (de type ARMA, ARMAX... [Hamilton 94]), soit des modélisations telles que les réseaux de neurones ou les arbres de décisions [Meek 02].

Plusieurs problèmes se font jour concernant le traitement de séries temporelles. Considérons par exemple un problème classique de traitement de séries temporelles

qui est la prédiction d'observations futures. Étant données les observations passées, que l'on notera $y_{1:t} = (y_1, ..., y_t)$ (dans cette thèse, nous ne traiterons que des observations à temps discret de telle sorte que t sera toujours un entier), déterminer la distribution de probabilité pour des observations futures $P(y_{t+h}|y_{1:t})$ ($h > 0$). La prédiction des observations futures, avec des modélisations dites "classiques", ne peut s'effectuer qu'en utilisant une fenêtre finie dans le passé $y_{t-l,t}$. Ainsi, pour des systèmes dont on ne connaît pas l'ordre (et dont l'ordre peut être supérieur à l), il y a perte d'information. De plus, de telles modélisations peuvent être inadaptées pour traiter des données multidimensionnelles. Enfin, il est compliqué d'intégrer des connaissances *a priori* dans de telles modélisations qui se basent seulement sur les données observables (sans structure sous-jacente).

Contrairement à ces méthodes "classiques", les modélisations à espaces d'états font l'hypothèse qu'il existe une structure sous-jacente. Cette structure, décomposée en états, génère les observations. Les modèles à espaces d'états présentent de nombreux avantages pour la modélisation de séries temporelles [Durbin 01], la modélisation sous-jacente permettant de s'affranchir des difficultés rencontrées par les approches classiques [Murphy 02]. Les modèles de Markov, modèles à espace d'états les plus répandus et étudiés, offrent notamment la possibilité d'intégrer une structure permettant de prendre en compte une connaissance *a priori* sur les observations.

Ce chapitre propose ainsi un état de l'art des principaux modèles de Markov cachés existants et de leurs utilisations. En effet, cette thèse se propose d'utiliser de telles modélisations afin de traiter des trajectoires issues de séquences d'images avec pour but des tâches de reconnaissance de contenus dans des vidéos.

1.2 Modèle de production des données

Un MMC est un automate à M états (la figure 1.1 présente un MMC à trois états), dont l'état à un instant t est noté s_t. Les paramètres $\lambda = (A, B, \pi)$ d'un MMC sont la matrice de transition A, le vecteur de distribution initiale π et, enfin, les probabilités d'observation conditionnelle B.

Ainsi, les probabilités de transition entre états sont données par la matrice A. La matrice de transition A est une matrice dite stochastique. En effet, la propriété des processus markoviens est que l'évolution de l'automate à l'instant $t + 1$ ne dépend que de la valeur de l'état de l'automate à l'instant t. La probabilité de passer d'un état m à un état m', notée $a_{mm'}$, est alors donnée par

$$a_{mm'} = p(s_t = m'|s_{t-1} = m),$$

avec

$$\sum_{m'=1}^{M} a_{mm'} = 1.$$

De plus, les MMC suivent une hypothèse d'homogénéité telle que, pour tout t,

$$p(s_t|s_{t-1}) = p(s_2|s_1).$$

Les probabilités que l'automate soit dans un état m initialement (*i.e.*, à $t = 0$) sont, elles, données par le vecteur π :

$$p(s_0 = m) = \pi(m).$$

Enfin, les probabilités d'observation conditionnelle permettant de lier les observations aux états du modèle sont données par $B = \{b_m(y_t)\},\ 1 \leq m \leq M$, où

$$b_m(y_t) = P(y_t \mid s_t = m).$$

Le caractère "caché" tient au fait que l'état du modèle n'est pas directement observé. De plus, l'observation est reliée à l'état au travers d'une variable aléatoire. C'est ce caractère aléatoire de la modélisation des observations qui, ajouté aux propriétés des processus markoviens, fait la souplesse et la puissance de ces approches.

Trois problèmes principaux peuvent se poser lors de l'utilisation d'un MMC :

- la **reconnaissance d'une séquence** : étant donnée une suite d'observations $y_{1:T}$ et un MMC, quelle est la séquence d'états $s_{1:T}$ sous-jacente la plus probable,

- la **probabilité d'observation d'une séquence** : étant donnée une suite d'observations $y_{1:T}$ et un MMC, quelle est la probabilité que cet automate ait engendré la séquence d'observations $y_{1:T}$,

- l'**apprentissage** : étant donnée une suite d'observations $y_{1:T}$, comment définir un MMC (au travers de ses paramètres) maximisant la probabilité d'observation de $y_{1:T}$ (*i.e.*, $p(y_{1:t}|\lambda)$).

Les algorithmes "standard" (*i.e.*, respectivement, les algorithmes de Viterbi, *forward-backward*, et de Baum-Welsh) permettant de traiter ces trois tâches sont présentés en annexes de ce document.

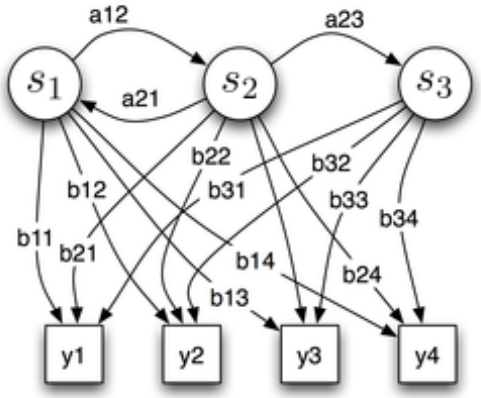

FIG. 1.1 – Représentation d'un MMC discret à 3 états.

1.3 Différents modèles de Markov cachés continus

Dans la suite de cette thèse, les valeurs considérées par les MMC sont des valeurs évoluant dans $\mathbb{R}$. Nous décrivons donc ici plusieurs modélisations markoviennes traitant des valeurs continues, parmi les nombreuses existantes. Le choix des modèles présentés ici a été fait en considérant les modèles markoviens qui apparaîtront dans la suite de ce manuscrit.

1.3.1 Modèles de Markov cachés continus "standards"

Lorsque les observations évoluent dans un espace continu (*i.e.*, $y_t \in \mathbb{R}^n$), il est courant de représenter les probabilités d'observations conditionnelles $p(y_t|s_t)$ à l'aide de gaussiennes :

$$b_i(y_t) = P(y_t = y|s_t = i) = \mathcal{N}(y; \mu_i, \Sigma_i),$$

où $\mathcal{N}(y; \mu, \Sigma)$ est la valeur d'une densité gaussienne de moyenne μ et de covariance Σ calculée en y ($y \in \mathbb{R}^k$) :

$$\mathcal{N}(y; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp((-\frac{1}{2}(y-\mu)'\Sigma^{-1}(y-\mu)).$$

Une représentation plus complète des observations peut être définie par des mélanges de gaussiennes :

$$b_i(y_t) = P(y_t = y|s_t = i) = \sum_{k=1}^{K_i} p(k_t = k|s_t = i)\mathcal{N}(y; \mu_{k,i}, \Sigma_{k,i}) = \sum_{k=1}^{K_i} c_{k,i}\,\mathcal{N}(y; \mu_{k,i}, \Sigma_{k,i}),$$

où k_t est une variable cachée désignant quel composant du mélange doit être considéré, et $p(k_t = k|s_t = i) = c_{k,i}$ est le poids conditionnel associé à chaque composante du mélange représentant l'état s_i.

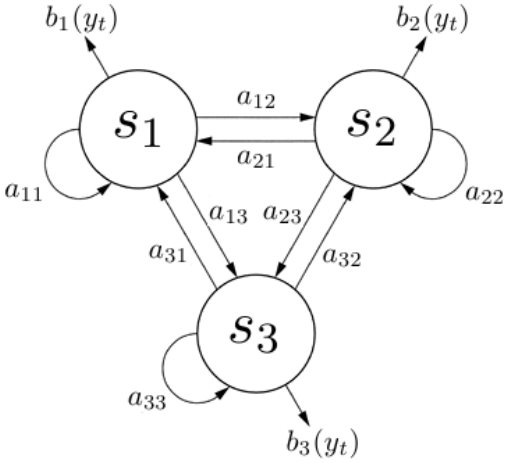

FIG. 1.2 – Représentation d'un MMC continu ergodique à 3 états.

Comme décrit dans la section B.3, l'algorithme de Baum-Welsh est applicable avec cette modélisation par mélange de gaussiennes. De tels modèles, utilisant des modélisations par mélanges de gaussiennes (notés MMG) seront maintenant dénotés par MMC/MMG (voir un exemple en figure 1.2).

Ici, B correspond donc aux paramètres des mélanges de gaussiennes, *i.e.*, les paramètres $c_{k,i}$, $\mu_{k,i}$ et $\Sigma_{k,i}$. Les formules permettant d'étendre l'algorithme de Baum-Welsh à ces paramètres sont données par [Rabiner 89] :

$$\bar{c}_{k,i} = \frac{\sum_{t=1}^{T} \gamma_t(k,i)}{\sum_{t=1}^{T}\sum_{k=1}^{K_i} \gamma_t(j,k)}, \bar{\mu}_{k,i} = \frac{\sum_{t=1}^{T} \gamma_t(k,i)}{\sum_{t=1}^{T} \gamma_t(k,i)} y_t,$$

$$\bar{\Sigma}_{k,i} = \frac{\sum_{t=1}^{T} \gamma_t(k,i)}{\sum_{t=1}^{T} \gamma_t(k,i)} (y_t - \mu_{k,i})(y_t - \mu_{k,i})'$$

où $'$ est l'opérateur de transposition et $\gamma_t(k,i)$ est la probabilité d'être dans l'état k à t avec la composante i du mélange de gaussiennes associée à l'état k, donnée par :

$$\gamma_t(k,i) = \left[\frac{\alpha_t(k)\beta_t(k)}{\sum_{l=1}^{M} \alpha_t(l)\beta_t(l)}\right]\left[\frac{c_{ki}\mathcal{N}(y_t, \mu_{ki}, \Sigma_{ki})}{\sum_{p=1}^{K_i} c_{kp}\mathcal{N}(y_t, \mu_{kp}, \Sigma_{kp})}\right].$$

$\gamma_t(k,i)$ est une version généralisée de $\gamma_t(k)$ dans l'équation B.2. En effet, $\gamma_t(k)$ est défini dans l'équation B.2 pour des représentations d'observations discrètes, et reste valable pour des représentations d'observations continues à l'aide de gaussiennes.

Pour modéliser des séquences d'observations temporelles, les MMC sont souvent utilisés avec une topologie dite *left-to-right* (ou de gauche à droite), telle que si $j \neq i$ et $j \neq i+1, p(s_j|s_i) = 0$. Fig. 1.3 présente un MMC *left-to-right* composé de 3 états. A l'inverse, si $\forall i, \forall j, p(s_j|s_i) \neq 0$, on dit que le MMC est ergodique. Certaines modélisations peuvent également adopter une topologie *left-to-right* "stricte", *i.e.* telle que si $j \neq i+1, p(s_j|s_i) = 0$.

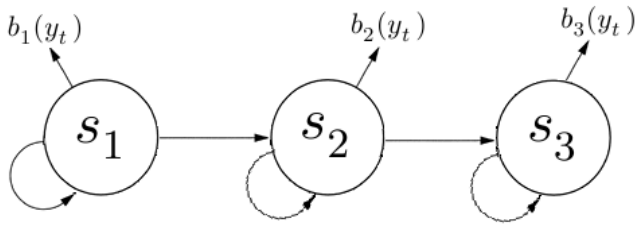

FIG. 1.3 – Un modèle de Markov de type *left-to-right* avec 3 états.

1.3.2 Modèles de Markov cachés Entrée/Sortie

Les MMC Entrée/Sortie (MMC E/S) sont des extensions des MMC continus "standards" introduits par Bengio et Frasconi [Bengio 95]. La différence est ici que les observations sont guidées par l'état caché s_t, mais également par l'observation x_t. Comme l'illustre la figure 1.4, la structure de base d'un MMC E/S est une structure de MMC continu "standard" *left-to-right* "stricte", *i.e.* telle que $j \neq i+1, p(s_j|s_i) = 0$.

Les MMC E/S construisent une séquence en sortie $\{y_t\}$ (qui ne correspond pas aux observations, mais par exemple dans [Just 04] pour des tâches de classification, à la classe estimée à t) en fonction d'une carte d'entrée et des paramètres du MMC E/S. Un MMC E/S est donc défini par des probabilités d'émission de la forme $p(y_t|s_t, x_t)$ et par des probabilités de transition de la forme $p(s_t|s_{t-1}, x_t)$. Les procédures d'entraînement des MMC E/S, correspondant à une adaptation de l'algorithme de Baum-Welsh à cette architecture, peuvent être trouvées dans [Bengio 95].

Au contraire des MMC, les MMC E/S sont dépendants de la dimension temporelle puisque les probabilités d'émission $p(y_t|s_t, x_t)$ et de transition $p(s_t|s_{t-1}, x_t)$ dépendent des observations x_t. À chaque état des MMC E/S sont associées deux distributions conditionnelles, une correspondant aux probabilités de transition et l'autre à la probabilité d'émission d'observation. Ainsi, la dynamique des systèmes n'est pas fixée *a priori* et évoluent dans le temps en fonction de la séquence d'observations.

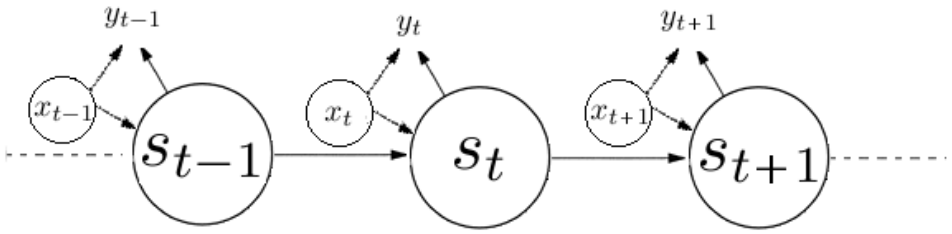

FIG. 1.4 – Un modèle de Markov de type Entrée/Sortie.

1.3.3 Modèles de Markov cachés couplés

Les MMC couplés sont utilisés lorsque l'on traite plusieurs sources d'observations de façon simultanée et que l'on veut prendre en compte de possibles interactions entre les sources d'observations.

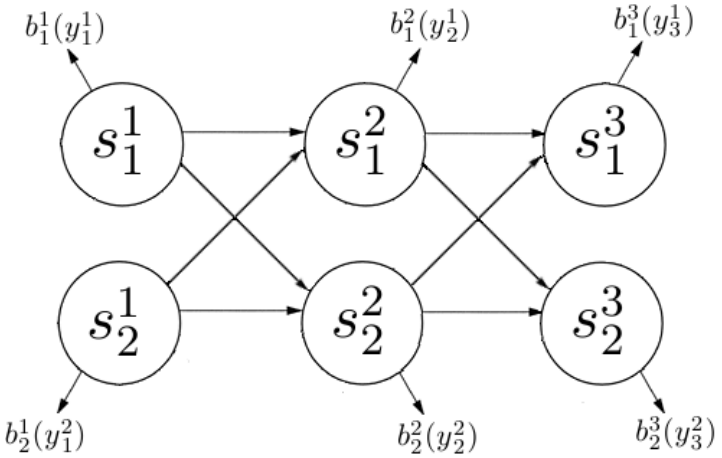

FIG. 1.5 – Un modèle de Markov caché couplé avec 2 chaînes "left-to-right" stricte.

Différentes architectures ont été proposées pour la modélisation des interactions entre processus à l'aide de MMC couplés. La figure 1.5 présente l'architecture la plus utilisée de MMC couplés. Cette modélisation comprend deux sources d'observations, une pour les observations $y_{1:T}^1$ et une pour les observations $y_{1:T}^2$, et permet de modéliser les interactions entre deux MMC de type *left-to-right* décrivant ces sources d'observations. Ici, les interactions sont telles que l'état à un instant t de l'un des deux MMC est fonction de l'état à l'instant $t-1$ des deux MMC. Les probabilités d'émission associées à ces modèles sont de la forme $p(x_{t+1}|x_t, y_t)$ et $p(y_{t+1}|x_t, y_t)$.

La principale problématique, avec les MMC couplés, est le temps de calcul important pour l'entraînement des modèles. Les temps de calcul, important avec deux processus (telle que celui présenté en figure 1.5), deviennent ingérables en considé-

rant plus de deux processus. Les détails sur les procédures d'inférence pour des MMC couplés pourront être trouvés dans [Brand 96].

1.3.4 Modèles de Markov cachés Parallèles

Les Modèles de Markov cachés Parallèles (MMCPa) sont une alternative aux MMC couplés pour le traitement de plusieurs processus simultanés. Dans le cas des MMC couplés discrets, la modélisation des interactions entre deux signaux de dimension C, chacune de ces dimensions décrite par un alphabet de K_i symboles, un total de combinaisons de l'ordre de $N = (\prod_i K_i)^2$ est à calculer. Le nombre N peut très vite s'avérer impossible à gérer avec des MMC couplés en termes de temps de calcul. Ainsi, une alternative aux MMC couplés sont les MMCPa (voir Figure 7.1.2.2).

L'hypothèse faite pour les MMCPa est simplement que l'ensemble des signaux traités évoluent de façon synchrone et indépendante. Ainsi, considérons les calculs de maximum de vraisemblance d'un MMCPa,

$$\max_{Q^1,\ldots,Q^C} \{\log P(Q^1,\ldots,Q^C,O^1,\ldots,O^C|\lambda_1,\ldots,\lambda_C)\},$$

où Q^i est la séquence d'état associée au signal i, dont la séquence d'observations est O^i et le MMC associé est décrit par λ_i. L'hypothèse d'indépendance des signaux permet d'avoir

$$\max_{Q^1,\ldots,Q^C} \{\log P(Q^1,\ldots,Q^C,O^1,\ldots,O^C|\lambda_1,\ldots,\lambda_C)\} = \max_{Q^1,\ldots,Q^C} \left\{ \sum_{i=1}^{C} \log P(Q^i,O^i|\lambda_i) \right\}.$$

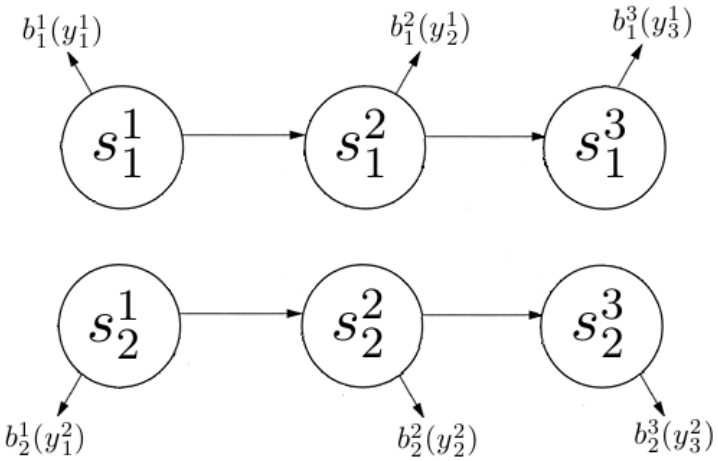

FIG. 1.6 – Un modèle de Markov caché parallèle avec 2 chaînes "left-to-right" stricte.

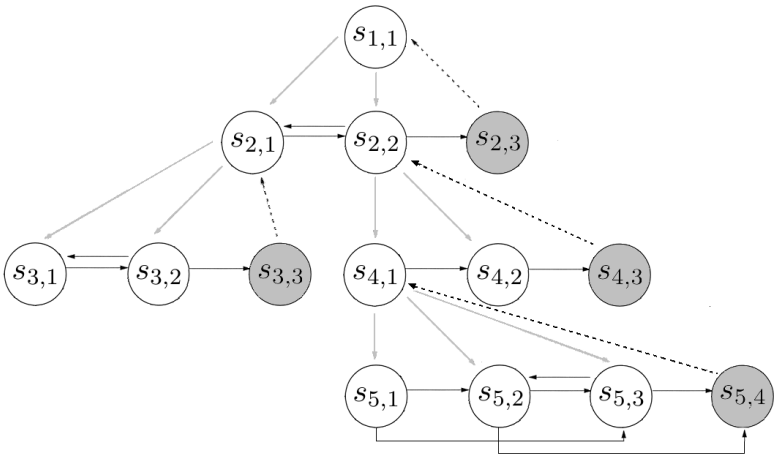

FIG. 1.7 – Un modèle de Markov hiérarchique. Les arcs gris correspondent à des transitions verticales descendantes, les arcs en pointillés représentent des transitions verticales ascendantes et les noirs des transitions horizontales. Les cercles clairs sont des états internes alors que les cercles foncés correspondent aux états terminaux. Pour des raisons de clarté, les états de production ne sont pas spécifiés.

1.3.5 Modèles de Markov cachés hiérarchiques

Dans un MMC hiérarchique, une hiérarchie est définie *a priori*. Lorsqu'un état d'un MMC hiérarchique est activé, un sous-MMC hiérarchique peut être appelé. Un des états du MMC sous-jacent est alors activé, qui peut lui même appeler un autre MMC hiérarchique. Ce processus est répété jusqu'à ce qu'un état, appelé état terminal, soit activé et alors le processus du MMC hiérarchique remonte à l'état du niveau supérieur concerné (transition signalée par les flèches en pointillés dans la figure 1.7). De plus, certains des états du MMC hiérarchique, appelés états de production, peuvent produire des données (des symboles par exemple). Sur la figure 1.7, ces états ne sont pas spécifiés pour des raisons de clarté.

Les méthodes permettant l'estimation des paramètres d'un MMC hiérarchique sont plus complexes que pour un MMC "standard". Le lecteur pourra se rapporter à [Fine 98] pour plus de détails. Il est à noter [Murphy 02] qu'un MMC hiérarchique peut toujours être interprété comme un MMC, et même comme un réseau bayésien dynamique.

Un MMC hiérarchique définit donc une structure *a priori* qui peut permettre de faciliter l'apprentissage. Pour les MMC "standard" par exemple, bien qu'un modèle de MMC ergodique puisse toujours être utilisé si suffisamment de données d'entraînement sont disponibles, il est courant de contraindre le modèle en empêchant certaines

transitions entre états (comme pour les MMC dits *left-to-right*). Ainsi et de la même manière, une structure de MMC hiérarchique doit permettre d'imposer une structure "forte" afin de résoudre plus efficacement, selon les données d'entraînement disponibles, les problèmes envisagés.

1.3.6 Modèles cachés semi-markoviens

Un modèle caché semi-markovien (que l'on notera dans la MCSM), également parfois appelé modèles segmental, est un MMC dans lequel on "force" les temps de séjour dans les états (c'est-à-dire le temps pendant lequel un processus reste dans un état) à suivre une distribution donnée.

En effet, les MMC tels qu'on les a définis jusqu'ici obligent les temps de séjour à suivre une distribution géométrique :

$$p(d_i) = a_{ii}^{d-1}(1 - a_{ii})$$

où d_i correspond au temps de séjour dans l'état i (voir Fig. 1.8).

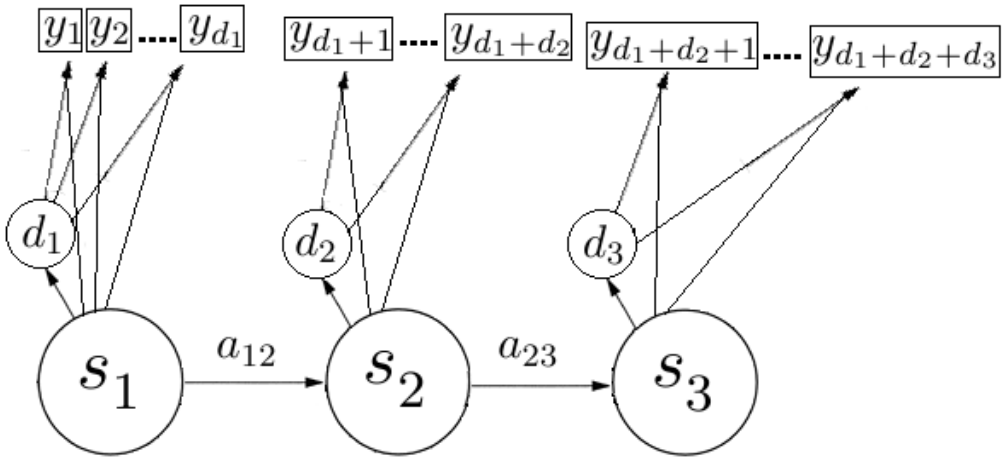

FIG. 1.8 – Un modèle de semi-Markov caché de type *left-to-right* avec 3 états.

Afin de modéliser le temps de séjour et de s'affranchir de cette distribution géométrique, d_i est modélisé comme une variable aléatoire définie pour chacun des états du MCSM, cette variable étant modélisée par un mélange de gaussiennes :

$$P(d_i) = \sum_{j=1}^{N_i} d_{j,i}\, \mathcal{N}(d_i; \mu_{j,i}, \Sigma_{j,i}),$$

où N_i correspond au nombre de composantes utilisées pour modéliser le temps de séjour de l'état i.

De même, les probabilités d'observation, avec les MCSM, peuvent être modélisées à l'aide de mélangee de gaussiennes calculés par des procédures de type *EM*. Dans [Ge 02], les probabilités d'observation des MCSM sont modélisées par des régressions linéaires.

Le processus d'un MCSM n'est en fait markovien qu'aux instants de changement d'états. Un segment étant ici défini comme une suite consécutives d'observations associées à un état donné ; supposons une séquence d'états à R segments, et soit q_r l'instant du r^{me} changement d'état, de telle sorte que le r^{me} segment correspond aux observations $y_{(q_{r-1}+1,q_r]} = y_{q_{r-1}+1}, \ldots, y_{q_r}$, avec $s_{q_{r-1}+1} = \ldots = s_{q_r}$. A est alors la matrice de transition du MCSM aux instants de changements d'états q_i et, donc, de transition entre segments :

$$p(s_{q_r} = j | s_{q_{r-1}} = i) = a_{ij},$$

et

$$\forall i, a_{ii} = 0.$$

⋆ Algorithme de Viterbi pour les MCSM

L'algorithme de Viterbi pour les MCSM est une version particulière de l'algorithme de Viterbi pour MMC, le processus n'étant markovien qu'au moment des changements d'états. Cet algorithme étant plus particulièrement utilisé dans la suite de ce manuscrit, nous proposons ici de détailler les étapes de cet algorithme de décodage.

Soit θ l'ensemble des paramètres du MCSM, *i.e.*, $\theta = \{A, \Pi, \{\theta_{d_i}\}, \{\theta_{obs_i}\}\}$ où A est la matrice de transition entre états S'_i, Π est le vecteur de distribution initiale des états, θ_{d_i} et θ_{obs_i} sont les paramètres respectifs des modélisations des temps de séjour et des observations.

Soit également une séquence d'observation y, l'algorithme de Viterbi pour les MCSM trouve la séquence d'états $\hat{S}$ du MCSM la plus probable, *i.e.*,

$$\begin{aligned} \hat{S} &= \arg\max_S P(S|y, \theta) \\ &= \arg\max_S P(S|y, \theta) P(y|\theta) \\ &= \arg\max_S P(S, y|\theta). \end{aligned}$$

Ainsi, pour chaque instant t et pour chaque état i, l'algorithme de Viterbi calcule $\hat{p}(i, t)$ définie par :

$$\hat{p}(i, t) = \max_{s_{[1,t]}} P(s_{[1,t]}, y_{[1,t]} | t = q_r, s_t = i),$$

où $s_{[1,t]}$ correspond à la suite d'états dans l'intervalle de temps $[1, t]$. $\hat{p}(i, t)$ peut être calculé de façon récursive, en effet on peut prouver que (les détails pouvant être trouvés dans [Ge 02]) :

$$\hat{p}(i,t) = \max_{t',j} \hat{p}(t',j) a_{ji} P(d_i = t - t' | \theta_{d_i}) P(y_{(t',t]} | \theta_{obs_i}).$$

Ainsi, l'algorithme de Viterbi pour les MCSM peut être décrit de la façon suivante, après initialisation de $\hat{p}(i, t = 0)$ à l'aide de Π, par la récursion suivante pour $t > 1$ et $1 \leq i \leq M$:

$$\hat{p}(i,t) = \max_{t'} \left(\max_{j} \left[\hat{p}(j,t') a_{ji} \right] P(d_i = t - t' | \theta_{d_i}) P\left(y_{(t',t]} | \theta_{obs_i} \right) \right). \tag{1.1}$$

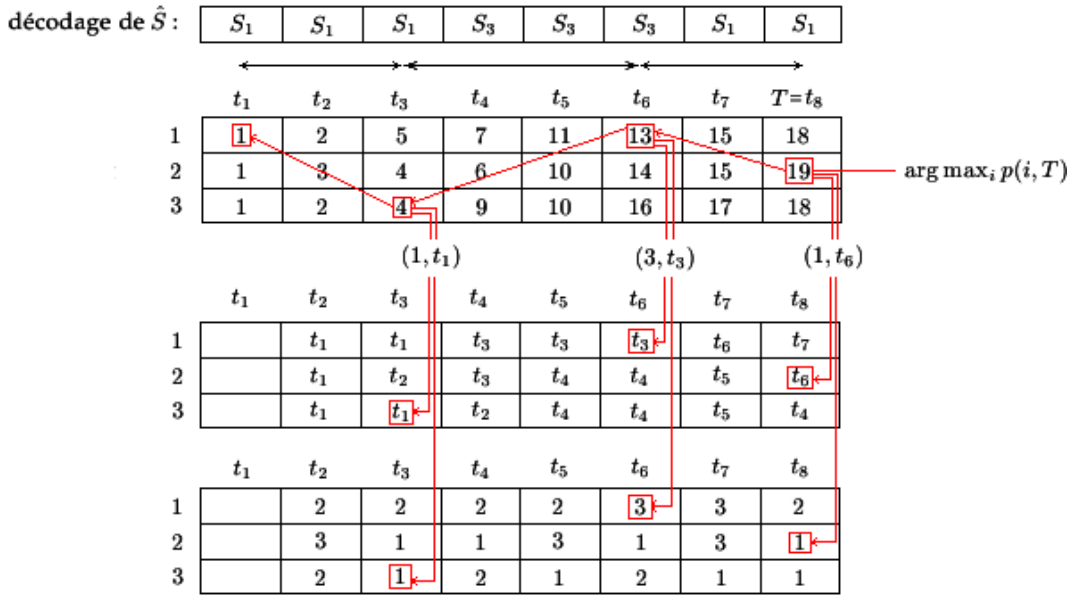

FIG. 1.9 – Illustration de l'algorithme de Viterbi pour les MCSM à l'aide d'un exemple simple, avec seulement trois états S_i'. Le premier tableau contient les valeurs de $p(i, t_k)$, i correspondant aux trois états du MCSM et t_k décrivant l'échantillonnage temporel. Le second tableau contient les temps de changement d'états précédents, le dernier tableau contenant lui les valeurs des états précédents. Ces deux derniers tableau sont utilisés pour le décodage de Viterbi pour "remonter", à partir de $\arg\max_i p(i,T)$, le temps afin de trouver la séquence optimale $\hat{S}$ d'états des MCSM.

Comme l'illustre la figure 1.9, les valeurs du temps t' et de l'état j maximisant cette dernière équation sont enregistrées dans deux tableaux (un pour enregistrer l'état précédent j et l'autre pour enregistrer le temps de changement d'état précédent t'). Alors, au couple (i,t) de l'équation 1.1, on associe le couple de changement d'état précédent (j,t'). Ainsi, après avoir trouvé l'état final k au temps final T, *i.e.* tel que $k = \arg\max_i p(i,T)$, il suffit, comme dans l'algorithme classique associé aux MMC, de

remonter le temps pour décoder la séquence d'états optimale. La différence avec l'algorithme de Viterbi pour les MMC étant que, plutôt que de remonter le temps instant par instant, on effectue ici des sauts (de t à t', et non de $t+1$ à t).

Pour plus de détails sur les algorithmes de Baum-Welsh pour les MCSM, le lecteur pourra se référer à [Ge 02].

1.4 Exemples d'utilisation des modèles de Markov cachés pour la reconnaissance de contenu

Nous présentons quelques applications en reconnaissance de contenu des modèles de Markov cachés. De nombreux autres travaux utilisant des modélisations markoviennes pour le traitement et l'utilisation de trajectoires peuvent également être trouvés dans le chapitre 3.

Les MMC ont été introduits par Baum et al. dans les années 60 [Baum 70], et ont tout d'abord été utilisés avec succès pour des tâches de reconnaissance de la parole [Rabiner 89]. Ils ont ensuite été largement utilisés dans différents domaines tels que, notamment, l'analyse de séquences biologiques (notamment de gènes, [Durbin 98]), de l'image [Li 00b] et de la vision par ordinateur [Starner 95]. s Ainsi, dans le domaine de la vision par ordinateur, les MMC ont pu être utilisés pour la reconnaissance de mouvements humains [Hoey 00, Sun 02, Dockstader 03, Cheng 08]. Ils ont aussi été largement considérés pour la reconnaissance de contenu dans des vidéos de sport [Kokaram 06, Barnard 05], par exemple pour des vidéos de football [Assfalg 02, Xie 04], de base-ball [Chang 02] ou de tennis [Kijak 03]. Les MMC ont également été appliqués à l'indexation vidéo multimodales, plus précisément en combinant des informations tant visuelles que sonores [Leonardi 02, Wang 00, Hung 07, Zhang 04] (voir plus précisément section 7.1.2.2).

Les autres types de modèles de Markov cachés ont également fait l'objet d'applications pour la reconnaissance de contenu dans différents domaines. Les MMC E/S ont eux notamment été utilisés pour la reconnaissance de mouvements 3D à l'aide de vidéos [Just 04] (voir plus de détails en section A). Des MMC couplés ont permis de traiter simultanément des données audio et vidéo dans des problèmes de reconnaissance de texte audiovisuelle [Nefian 02]. Ils ont également été appliqués à la reconnaissance de comportements dans des vidéos par Oliver et al. [Oliver 00, Brand 96]. Les tâches de reconnaissance de la parole ([Bourlard 97]) et en reconnaissance du langage des signes dans des vidéos ([Vogler 99]) ont pu être réalisées à l'aide de MMCPa. Les MMC hiérarchiques ont été notamment exploités avec succès dans des tâches de reconnaissance d'écriture [Fine 98] ou de reconnaissance dans des vidéos de football [Xie 03]. Enfin, dans [Ge 02], les MCSM ont été utilisés pour la reconnaissance, notamment, de textes et de séquences biomédicales, alors que Natarajan et al. [Natarajan 07a,

Natarajan 07b] en ont fait usage pour des tâches de reconnaissance d'activités dans des vidéos.

Ces quelques exemples, parmi les nombreuses applications des MMC en vision par ordinateur, montrent la variété des utilisations des MMC pour des tâches de reconnaissance de contenus, et plus particulièrement dans le domaine des vidéos de sport, domaine d'applications que nous considérerons dans la suite.

1.5 Conclusion

Nous avons fait dans cette première partie un état de l'art des principaux modèles de Markov cachés existants et utilisés pour la reconnaissance de formes. Les modélisations markoviennes cachées sont, comme on a pu le voir dans ce chapitre, des outils largement utilisés pour la reconnaissance de contenus et spécialement dans le domaine de la vision par ordinateur. Fondés sur des bases mathématiques solides, ces modèles possèdent des techniques d'apprentissage automatique efficaces. Du fait de leurs diverses propriétés, notamment d'exploitation des causalités temporelles, de prise en compte de l'incertitude attachée aux données et d'efficacité de mise en oeuvre, les MMC ont été largement privilégiés. Ils permettent aussi de définir des modélisations dédiées (selon les types de MMC utilisés) aux problématiques envisagées.

En effet, les différentes modélisations markoviennes permettent de mettre en place des architectures spécifiques aux problèmes rencontrés. Les avantages sont, pour chaque types de MMC présentés, les suivants :

- les MMC permettent de modéliser les variabilités temporelles inhérentes aux observations rencontrées, et proposent plusieurs algorithmes élégants et dont l'efficacité à été prouvée, tant théoriquement qu'expérimentalement.
- les MMC E/S offre, au contraire des MMC, une flexibilité à la dimension temporelle. La dynamique des systèmes de ces modèles n'est en effet pas fixée *a priori* et évoluent dans le temps en fonction des observations.
- Les MMC couplés peuvent être utilisés pour traiter plusieurs sources d'observations de façon simultanée tout en prenant en compte leurs possibles interactions.
- les MMC Pa, eux, constituent une alternative aux MMC couplés pour le traitement de plusieurs processus simultanés. Ils permettent de pallier aux problèmes de temps de calcul pouvant être rencontrés par les MMC couplés mais ne prennent pas en compte les interactions entre les différents processus observés.
- Une structure *a priori*, permettant de prendre en compte précisément des connaissances *a priori* sur les données observées, peut être mise en place à l'aide des MMC hiérarchiques. Cette architecture impose une structure "forte" et contraignante, et peut permettre de faciliter l'apprentissage des modèles.
- Les MCSM offrent une prise en compte des temps de séjour dans les états plus précise que les MMC. Cette propriété peut aider, pour certains problèmes, à dé-

finir des modélisations plus précises et plus efficace que les MMC.
Ainsi, dans la suite de ce document, et en fonction des problèmes envisagés, différents types de modélisations markoviennes seront utilisées et proposées pour le traitement de trajectoires. Les MMC doivent en effet aider à la prise en compte des causalités temporelles inhérentes aux trajectoires vidéos exploitées dans notre travail.

Les modélisations décrites dans ce chapitre et plus précisément les algorithmes correspondants nécessitent des ensembles de données de tailles importantes. Des tâches de reconnaissance rencontrées dans la suite nous amèneront notamment à mettre en place de MMC dédiés au traitement de petits ensembles de données. De plus, des modélisations markoviennes combinant MCSM, MMC hiérarchiques et MMC Pa seront également développées pour des problèmes de segmentation temporelle de vidéos. Le présent chapitre (ainsi que les algorithmes présentés en annexes) est donc une base théorique permettant de mettre en perspective les parties suivantes de cette thèse.

Chapitre 2

Algorithme de classification non-supervisée (ou clustering)

"Or, si l'on décide de complémentariser la notion d'organisation et celle d'organisme, si la première n'est pas strictement réductrice, analytique, mécanistique, si la seconde n'est pas seulement totalité porteuse d'un mystère vital indicible, alors on peut approcher un peu plus le problème du vivant. Car c'est bien avec la vie que la notion d'organisation prend une épaisseur organismique, un mystère romantique. C'est là où apparaissent des traits fondamentaux inexistants dans les machines artificielles : une relation nouvelle par rapport à l'entropie, c'est-à-dire une aptitude, ne serait-ce que temporaire, à créer de la neguentropie, à partir de l'entropie elle-même ; une logique beaucoup plus complexe et sans doute différente de celle de toute machine artificielle. Enfin, lié indissolublement aux deux traits que nous venons d'énoncer, il y a le phénomène de l'auto-organisation."

Edgar Morin - ***Introduction à la pensée complexe***

2.1 Introduction

Ce chapitre propose un court état de l'art des principaux outils mathématiques de classification non-supervisée des données. En effet, cette thèse se propose d'utiliser des modélisations markoviennes (voir le chapitre précédent) de trajectoires issues de

séquences d'images avec pour but, notamment, des tâches de reconnaissance non-supervisée de contenus dans des vidéo. Ainsi, ce chapitre sera consacré aux méthodes existantes pour la classification non-supervisée de données.

Cet état de l'art ne se veut pas exhaustif et se limite aux algorithmes les plus connus et les plus utilisés. Une étude plus approfondie des algorithmes de clustering pourra se trouver dans [Jain 99, Grabmeier 02]. Il a été fait le choix de diviser les algorithmes de clustering en trois parties distinctes : les approches hiérarchiques, les approches d'estimation paramétrique de la densité et enfin les approches d'estimation non paramétrique de la densité. Il est à noter que certaines des méthodes décrites peuvent appartenir à plusieurs des catégories définies. Dans la suite de ce manuscrit, des algorithmes de clustering appartenant à chacune des trois classes seront considérés et utilisés dans la suite de ce manuscrit.

Dans la suite de cette section, les notations seront les suivantes : soit u un cluster et K le nombre de clusters. Les individus (ou points de l'espace des données) à classer sont notés x_i, sont au nombre de I, et $\mathcal{C}_u$ désigne l'ensemble des individus appartenant au cluster u. La fonction $c(.)$ est la fonction d'appartenance des individus aux clusters, *i.e.* telle que :

$$c(i) = u \Leftrightarrow x_i \in \mathcal{C}_u.$$

2.2 Approches hiérarchiques

Le principe des approches de clustering hiérarchique est de construire un arbre binaire appelé dendogramme (la figure 2.1 présente un dendogramme). Au niveau le plus bas, chaque individu constitue un cluster, puis niveau après niveau, on regroupe les clusters les plus proches en utilisant une mesure de similarité entre clusters définie *a priori*.

Le clustering des individus se fait en coupant l'arbre horizontalement (voir la droite orange dans la figure 2.1), chaque intersection entre la "limite" horizontale et les branches du dendogramme définit un cluster composé des individus (ou feuille de l'arbre binaire) composant la branche. Par exemple, dans la figure 2.1, la ligne orange intersecte quatre branches, il y a donc quatre clusters composés des individus de chaque branche (respectivement les individus de la branche verte, bleue, rouge et violette). La construction des arbres peut se faire de façon ascendante [Lance 67] ou descendante [Hubert 74]. Afin de procéder au clustering final, deux choix sont possibles, le premier étant de donner *a priori* le nombre k_{clust} de clusters désirés, le second étant de définir une valeur de similarité comme seuil d'arrêt, les regroupement entre clusters s'arrêtant dès lors que ce seuil est atteint.

Le principal avantage des méthodes hiérarchiques de clustering est la stabilité des résultats, meilleure que pour les autres méthodes de clustering, le principal inconvé-

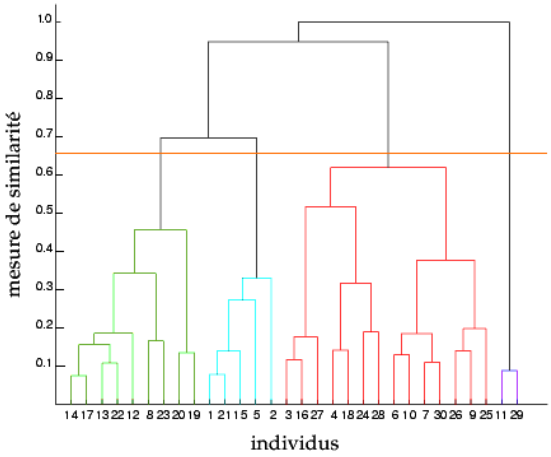

FIG. 2.1 – Exemple de dendogramme produit par une méthode de clustering hiérarchique ascendante, avec en abscisse les individus dont on veut former des clusters et en ordonnées les valeurs de mesure de similarité correspondant aux unions entre clusters.

nient étant le coût de calcul qui interdit l'utilisation de ces méthodes pour de grands ensembles d'individus. Ces algorithmes de clustering, de par la stabilité offerte en termes de résultats, seront utilisés pour les tâches de clustering entre individus (et notamment entre trajectoires) dans la suite de cette thèse.

2.3 Approches d'estimation paramétriques de la densité

Le but de ces méthodes est, comme leur nom l'indique, d'estimer la densité des individus à l'aide d'un modèle statistique défini *a priori*. L'objectif est généralement d'estimer les paramètres d'un modèle de mélange de distributions de la forme :

$$f(x) = \sum_{i=1}^{K} p_i \phi_i(x, \lambda_i),$$

où ϕ est une distribution choisie, p_i correspond au poids associé à ϕ_i et λ_i est l'ensemble des paramètres de la distribution ϕ_i.

Le choix le plus fréquent est de considérer des distributions gaussiennes (voir figure 2.2). Les paramètres λ correspondent alors aux moyennes et matrices de covariance des gaussiennes obtenues à l'aide d'un algorithme de type *Expectation-Maximization* (*EM*, dont l'algorithme de Baum-Welsh présenté en section B.3 est un cas particulier pour les MMC)[Dempster 77]. Le lecteur intéressé pour l'application de l'algorithme *EM* à des mélanges de gaussiennes pourra se référer à [Bilmes 98].

Les résultats des approches d'estimation paramétrique de la densité des individus dépendent de l'adéquation de la distribution "réelle" des individus avec les formes de distributions choisies, et, également, de l'initialisation du modèle. Il est également à noter que l'algorithme *EM* peut s'avérer assez coûteux.

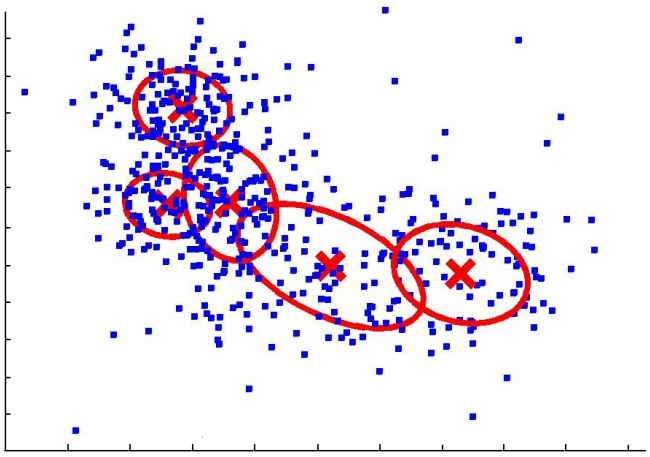

FIG. 2.2 – Exemple de clustering obtenu par l'algorithme *EM* sur des données en deux dimensions. Les croix rouges indiquent les moyennes des gaussiennes et les enveloppes rouges les covariances, les individus étant les points bleus.

• **Choix du nombre de composantes des mélanges et algorithmes EM**

Une problématique importante de l'algorithme *EM* est de choisir le nombre de composantes (ou modes) d'un mélange. Plusieurs critères ont été proposés, les deux critères les plus utilisés étant le "Bayesian Information Criteria" (*BIC*, [Schwarz 78]), le critère "Akaike Information Criteria" (*AIC*, [Akaike 73]). Pour une étude plus précise de ces critères, le lecteur pourra se référer à [Lebarbier 04]. Le principal problème de ces critères est qu'ils ont tendance à sur-représenter les données et, donc, à surestimer le nombre de composantes des mélanges. Un critère inspiré du critère *BIC* et pouvant permettre, dans une certaine mesure, de s'affranchir de cet inconvénient est le critère *ICL* (pour "integrated completed likelihood") proposé par Biernacki et al. [Biernacki 00].

Les algorithmes de clustering paramétrique, et notamment l'algorithme *EM* seront considérés dans la suite de ce travail lors des utilisations de MMC, et notamment de MMC utilisant des gaussiennes pour la modélisation de données continues.

2.4 Approches d'estimation non paramétriques de la densité

La dernière classe d'algorithmes de clustering correspond aux approches d'estimation non paramétriques de la densité. Contrairement aux approches d'estimation paramétrique décrites dans la section précédente, la forme des distributions des individus à considérer n'est pas imposée.

L'objectif de certaines méthodes de clustering par partitionnement des données non paramétrique est de créer l'ensemble de clusters d'individus x_i tel que S soit minimisé, où S est défini par :

$$S = \sum_{u=1}^{K} \sum_{i|c(i)=u} d(x_i, v_u),$$

d correspondant à une mesure de similarité entre individus, et v_u étant le représentant du cluster u (souvent choisi comme étant le centre de gravité du cluster u). Dans ces méthodes, le nombre de clusters K doit être défini *a priori*, ainsi qu'une partition initiale des données, cette partition initiale étant par la suite améliorée de façon itérative.

Le principal inconvénient de ces méthodes est la sensibilité aux conditions initiales, qui peuvent mener l'algorithme vers des minima locaux.

L'algorithme de partitionnement des données non paramétrique de ce type le plus connu est l'algorithme dit *k-means* (voir figure 2.3). Dans cet algorithme, d est la distance euclidienne, et l'optimisation de S est effectuée par descente de gradient. Quelques applications (parmi les nombreuses existantes) de différentes version de l'algorithme *k-means* peuvent être soulignées, en segmentation d'images [Kanungo 02], en clustering d'image [Charalampidis 05] ou en clustering de vidéos [Maliatski 05].

Deux autres algorithmes de clustering non paramétriques connus sont le *mean shift* (initialement introduit par [Fukunaga 75]) et le *graph cut*. Le *mean shift* est une procédure itérative de montée du gradient permettant l'estimation des modes de la densité d'un nuage d'individus. Les aspects théoriques de l'algorithme *mean shift*, ainsi que son application notamment en segmentation d'images peuvent être trouvés dans [Comaniciu 02]. La méthode de *graph cut* est un algorithme de "coupe minimale - flot maximal" effectué dans un graphe reliant les individus, les arcs entre individus correspondant aux similarités entre individus [Boykov 01]. Les algorithmes *graph cut* ont été appliqués pour la première fois, en traitement de l'image, par Greig et al. [Greig 89]. Ils ont également été utilisés par Bugeau et Pérez pour la segmentation d'objets en mouvement dans des vidéos [Bugeau 08]

Les avantages de ces algorithmes sont qu'ils convergent rapidement et ne demande pas d'*a priori* sur les données. De plus, il ne rencontre pas de problèmes de convergence.

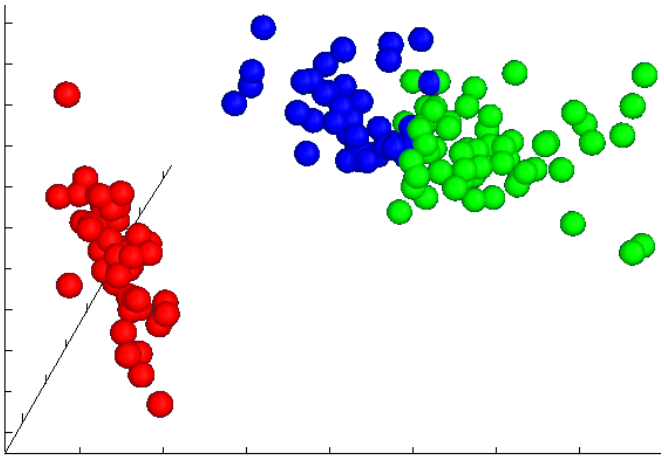

FIG. 2.3 – Exemple de clustering obtenu par l'algorithme *k-means* sur des données en trois dimensions.

Une dernière sorte d'algorithmes largement utilisés pour le clustering non paramétrique de données sont les algorithmes de clustering spectral. De tels algorithmes effectuent un clustering d'individus en utilisant les vecteurs propres de matrices dérivées des données (pour une étude sur le clustering spectral et son utilisation, voir [Weiss 99]). Un algorithme de clustering spectral a été proposé par Ng et al. [Ng 02] et se décompose comme suit,

Soit une suite de n individus $S = \{s_1, \ldots, s_n\}$ dans $\mathbb{R}^l$ et k le nombre de clusters désiré :

1. Construire la matrice d'affinité $A \in \mathbb{R}^{n \times n}$ telle que $A_{ij} = \exp(- \parallel s_i - s_j \parallel^2 /2\sigma^2)$ si $i \neq j$ et $A_{ii} = 0$.
2. Construire D la matrice diagonale telle que $D_{ii} = \sum_{j=1}^{n} A_{ij}$ et L telle que $L = D^{-1/2} A D^{-1/2}$.
3. Calculer $x_1, x_2, \ldots, x_k$ les k plus grands vecteurs propres de L (choisis orthogonaux si une même valeur propre correspond à un espace de dimension supérieure à 1), et construire la matrice $X = [x_1 x_2 \ldots x_k] \in \mathbb{R}^{n \times k}$.
4. Construire Y la version normalisée de X, *i.e.* telle que $Y_{ij} = X_{ij}/(\sum_j X_{ij}^2)^{1/2}$.
5. Effectuer un clustering en k groupes des lignes de Y à l'aide de *k-means*.
6. Assigner l'individu s_i au cluster j si la ligne i de Y est assignée au cluster j.

Il est à noter que les algorithmes spectraux nécessitent l'appel à un autre type d'algorithme de clustering non paramétrique (ici l'algorithme *k-means*), et sont donc concernés par les inconvénients de ces algorithmes. De plus, la sélection automatique du nombre de clusters k et le choix des vecteurs propres à considérer pour un cluste-

ring optimal restent des problèmes ouverts. Il est également à souligner la contrainte que les individus initiaux doivent avoir une taille égale.

Une application intéressante de l'algorithme de clustering spectral à la segmentation d'image a été proposée par Shi et al. [Shi 00].

Dans la suite de la présentation de ce travail de thèse, l'algorithme non-paramétrique de clustering *k-means* sera utilisé notamment pour l'initialisation d'algorithme d'estimation paramétrique de la densité de type *EM*.

2.5 Conclusion

Nous avons donc fourni un court état de l'art des méthodes existantes pour le clustering de données. La classification non-supervisée de données est un vaste domaine et nous avons ici présenté les principales alternatives utilisées en trois parties distinctes, des méthodes de chacune d'entre elles étant exploitées dans la suite de ce document.

Chapitre 3

État de l'art des méthodes de traitement des séries temporelles et des trajectoires vidéos

---

"L'information suppose la computation vivante. De plus, je dois faire cette précision : la computation ne se ramène nullement au traitement des informations. La computation vivante comporte à mes yeux une dimension non digitale. [...] La bactérie ne connaît pas ce qu'elle connaît, et elle ne sait pas qu'elle sait. L'appareil cérébral des animaux constitue un appareil différencié de la connaissance. Il ne compute pas directement les stimuli que trient et codent les récepteurs sensoriels ; il compute les computations que font ses neurones.
Apparaît alors la différence entre information et connaissance, car la connaissance est organisatrice. La connaissance suppose un rapport d'ouverture et de fermeture entre le connaissant et le connu. [...] C'est le problème du computo-auto-exo-référent. C'est le problème de la frontière qui isole la cellule et qui en même temps la fait communiquer avec l'extérieur. Le problème, c'est de concevoir l'ouverture qui conditionne la fermeture et vice versa."

Edgar Morin - ***Introduction à la pensée complexe***

Ce chapitre propose un état de l'art des travaux développés dans le domaine de la reconnaissance de contenu vidéo. La reconnaissance de contenu dans des vidéos est un large domaine de la vision par ordinateur. Ainsi, ce chapitre présente un aspect

spécifique de la reconnaissance de contenu dans les vidéo, correspondant au cadre considéré dans cette thèse : l'analyse et le traitement des séries temporelles et des trajectoires issues de vidéos.

Les trajectoires étudiées dans cette thèse sont formées par des suites de coordonnées (une par image et par objet suivi) dans le plan vidéo. Ainsi, l'analyse de trajectoires est à envisager du point de vue du traitement de séries temporelles (*i.e.*, des séquences de valeurs ordonnées).

Dans cette partie de l'état de l'art, les méthodes existantes et d'intérêt pour l'analyse de séries temporelles sont tout d'abord décrites. Ce domaine d'analyse des séries temporelles permet une introduction à la présentation des méthodes utilisées plus spécifiquement pour l'analyse de trajectoires vidéos, objet de la seconde section de ce dernier chapitre de l'état de l'art.

3.1 Analyse de séries temporelles

L'état de l'art des méthodes permettant de traiter des séries temporelles est ici divisée en deux parties. Comme le montre la figure 3.1, la première partie sera consacrée aux méthodes de traitement de séries temporelles basées sur les données (avec et sans pré-traitement des données), alors que la seconde partie sera dédiée aux modélisations statistiques faites des séries temporelles de données, approches qui seront plus spécifiquement utilisées dans la suite du document (pour une étude sur le clustering de séries temporelles : [Liao 05]).

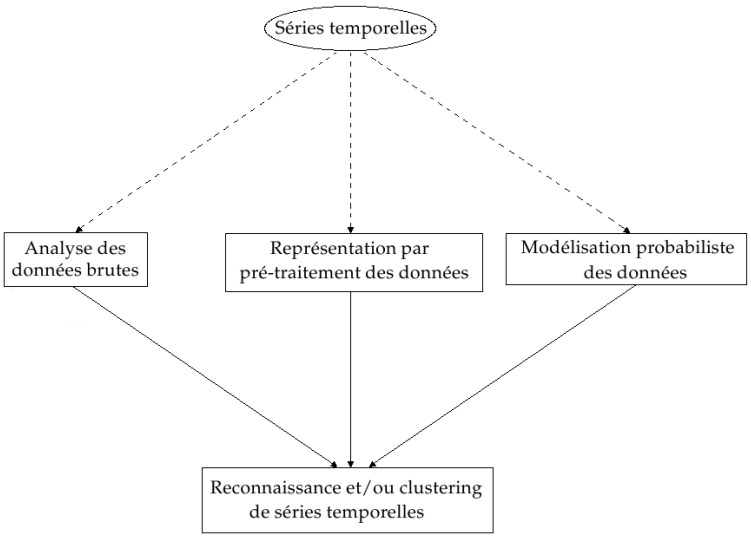

FIG. 3.1 – Présentation générale des méthodes de traitement des séries temporelles.

3.1.1 Approches de traitement de séries temporelles déterministes

Cette partie s'intéresse aux approches qui traitent directement les séries de données sans mettre en place de modélisations *a priori*. Les méthodes présentées ici pour la reconnaissance ou le clustering de séries temporelles sont donc largement basées sur l'utilisation de distances entre les géométries des séries temporelles considérées. Le but n'est pas de donner une liste exhaustive des travaux réalisés pour l'analyse de séries temporelles, ce qui pourrait s'avérer fastidieux et n'aurait pas un intérêt décisif dans ce manuscrit. L'objectif est plutôt de décrire de façon ordonnée les principales voies existantes et les travaux qui nous paraissent les plus intéressants.

Nous avons donc décidé de différencier deux types d'approches regroupant les méthodes développées pour la comparaison et l'analyse déterministe de séries temporelles, tout d'abord les méthodes traitant les suites de valeurs "directement" puis celles effectuant un pré-traitement des données pouvant, par exemple, correspondre au passage dans un nouvel espace de représentation des données.

3.1.1.1 Approches de traitement de séries temporelles sur les données brutes

Les distances entre séries temporelles sont décrites pour des données temporelles de dimension 1 et ce pour des raisons de simplicité de présentation, elles s'étendent toutes trivialement à des données temporelles de dimension supérieure.

⋆ ***Distances de Minkowski***

Ces approches s'appuient essentiellement sur l'utilisation de distances entre séries temporelles. La première approche, la plus simple, est de considérer la distance euclidienne entre séries temporelles. Soit deux séries temporelles v and w de la même taille N. La distance euclidienne entre v et w est définie par :

$$d_{Eucl}(v,w) = \sqrt{\sum_{k=1}^{N}(v_i - w_i)^2}.$$

La distance géométrique moyenne, très proche de la distance euclidienne, est définie par :

$$d_{gm}(v,w) = \frac{d_{Eucl}(v,w)}{N}.$$

La distance euclidienne étant elle-même un cas particulier de la distance de Minkowski définie par :

$$d_{Mink}(v,w) = \sqrt[p]{\sum_{k=1}^{N}|v_i - w_i|^p}.$$

Il est à noter que ces distances ne peuvent comparer que des séries temporelles de même taille, ce qui s'avère très restrictif. De plus, cette distance est sensible aux distorsions temporelles. Parmi les travaux utilisant la distance géométrique moyenne pour le traitement de séries temporelles et plus particulièrement des tâches de clustering, on peut citer [Van Wijk 99].

⋆ Distance "Dynamic Time Warping" (DTW)

La distance dite *Dynamic Time Warping*, proposée initialement par Myers et al. [Myers 81], est largement exploitée dans la comparaison de séries temporelles. Soit deux séries temporelles $v = v_1, \ldots, v_N$ et $w = w_1, \ldots, w_M$ de tailles N et M. On définit la fonction $H(v)$ par $H(v) = v'$ où $v' = v_1, \ldots, v_{N-1}$. La distance *DTW* entre v et w est définie de façon récursive par [Berndt 94] :

$$DTW(v,w) = d_{Eucl}(v_N, w_M) + \min\big(DTW(H(v),H(w)), DTW(H(v),w), DTW(v,H(w))\big).$$

Le principal avantage de la *DTW* est qu'elle permet de comparer des séries temporelles tout en tolérant une déformation temporelle des observations, ce qui lui confère une souplesse et une flexibilité dans la comparaison de séries temporelles par rapport, par exemple, aux distances de Minkowski, qui comparent les valeurs temporelles "point à point". La *DTW* permet donc de comparer des séries temporelles de tailles différentes, en effectuant les comparaisons à l'aide d'un alignement temporel des données. La distance *DTW* a été utilisée avec succès notamment en recherche de données [Keogh 00, Yi 98], en reconnaissance de gestes [Gavrila 95] ou en reconnaissance de parole [Rabiner 93]. Une limite inférieure permettant de fortement diminuer les temps de calcul pour l'indexation de séries temporelles a été proposée dans [Keogh 02]. Un développement intéressant de la *DTW* appelé *Derivative Dynamic Time Warping* peut être trouvé dans [Keogh 01].

⋆ Distance "Longest common subsequence" (LCSS)

La distance s'appuyant sur la *LCSS* (que l'on notera abusivement distance *LCSS* dans la suite) est une distance efficace pour la comparaison de séries temporelles, et notamment en présence de bruit. Soit deux séries temporelles $v = v_1, \ldots, v_N$ et $w = w_1, \ldots, w_M$ de tailles N et M. On définit, de la même manière que pour la distance *DTW*, la fonction $H(v)$ par $H(v) = v'$ où $v' = v_1, \ldots, v_{N-1}$. Avant de décrire la distance $LCSS$, on définit :

$$LCSS_{\delta,\epsilon}(v,w) = \begin{cases} 0 & \text{si } v \text{ ou } w \text{ est vide} \\ 1 + LCSS_{\delta,\epsilon}(H(v),H(w)) & \text{si } |v_N - w_M| < \epsilon \text{ et } |N-M| \leq \delta \\ \max(LCSS_{\delta,\epsilon}(H(v),w), LCSS_{\delta,\epsilon}(v,H(w))) & \text{sinon} \end{cases}$$

Le paramètre δ est un entier et ϵ est tel que $0 < \epsilon < 1$. La distance *LCSS* finalement considérée est :

$$D_{\delta,\epsilon}(v, w) = 1 - \frac{LCSS_{\delta,\epsilon}(v, w)}{\min(N, M)}.$$

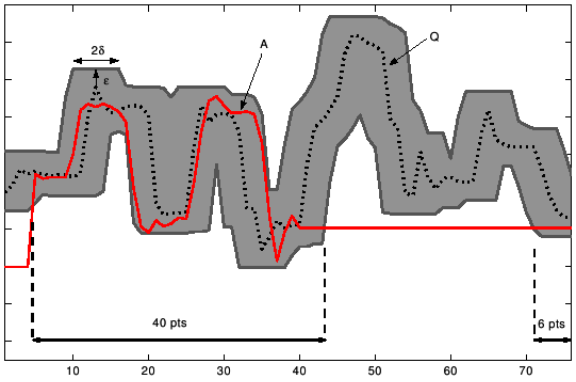

FIG. 3.2 – Illustration de la distance *LCSS*, en gris l'enveloppe définie autour de la série temporelle (en pointillés) par les paramètres δ et ϵ, et en rouge une série temporelle pour comparaison.

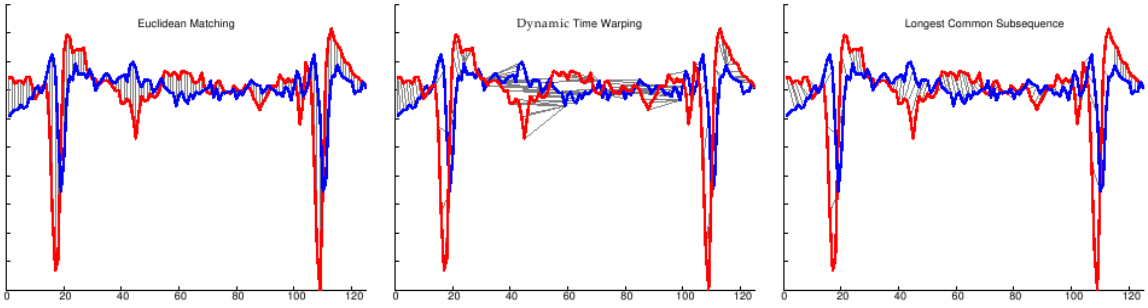

FIG. 3.3 – Un exemple de l'association faite par les 3 distances présentées entre les points de deux séries temporelles (une en rouge et une en bleu).

Le principal inconvénient de cette méthode réside dans le choix des paramètres δ et ϵ (voir Fig. 3.2, tirée de [Vlachos 03], pour une illustration du rôle de δ et ϵ). A notre connaissance, il n'existe pas de moyen justifié de choisir automatiquement ces deux paramètres.

Il a été montré [Vlachos 03, Vlachos 06] que cette distance permet une comparaison de séries temporelles efficace, potentiellement meilleure que la distance *DTW* pour des choix de δ et ϵ idoines, et ce notamment en présence de bruit (voir Fig. 3.3, tirée de [Vlachos 03], qui illustre les associations point à point pour les trois méthodes

décrites). Elle a notamment été utilisée pour la comparaison de trajectoires vidéos [Vlachos 02, Vlachos 02b, Buzan 04] et sera considérée comme élément de comparaison pour certains problèmes dans la suite de ce manuscrit.

3.1.1.2 Analyses de séries temporelles par pré-traitement des données brutes

L'intérêt d'effectuer un pré-traitement réside souvent dans la réduction qui est faite de la dimension des espaces de données. Les méthodes présentées ci-dessus exploitant directement les données brutes nécessitent de travailler dans des espaces potentiellement de grande dimension. Ces réductions de dimension peuvent permettre de diminuer le temps de calcul, ainsi que la probabilité de certains algorithmes de clustering (de type *EM* ou *k-means*) de converger vers des minima locaux [Ding 02]. De plus, les données pouvant être fortement bruitées, un pré-traitement peut s'avérer nécessaire et efficace pour l'analyse de séries temporelles.

Des premières tentatives de pré-traitement de données à l'aide de représentation par transformée de Fourier [Agrawal 93, Faloutsos 94] ou par décomposition en valeurs singulières [Korn 97] ont été proposées. Ces représentations ont ensuite été surpassées par les représentations à l'aide d'ondelettes [Wu 00].

Les transformées en ondelettes ont donc été exploitées pour l'indexation et le clustering de séries temporelles [Popivanov 02]. Les ondelettes ont en effet largement été utilisées pour représenter les signaux de façon compacte [Mallat 99]. Ces ondelettes forment une base de fonctions de base de variation multi-échelles, ou multi-résolution, utilisées dans le but de l'approximation et/ou de la compression des données. Les transformées en ondelettes de Haar ont également été considérées pour l'analyse de séries temporelles [Chan 99, Vlachos 03b, Lin 04]. La décomposition en ondelettes de Haar se déroule niveau par niveau, chaque niveau de transformation permettant de représenter plus précisément le signal. Vlachos et al. ont donc exploité ces différents niveaux de représentation, et ont proposé un algorithme appelé *I-k-means* (pour *iterative k-means*) qui s'appuie sur l'algorithme de clustering *k-means* (voir section 2.4). A chaque niveau de représentation en transformée de Haar, l'algorithme effectue un traitement par *k-means* de partitionnement des données, partitionnement ensuite utilisé au niveau suivant de représentation comme initialisation et ainsi de suite jusqu'à ce qu'il n'y ait plus de changement dans le partitionnement de données obtenu.

L'algorithme *k-means* a également été appliqué par Wilpon et al. [Wilpon 85] pour la reconnaissance de mots. Dans ce travail, les coefficients de Codage Prédicatif Linéaire (*CPL*) ont été considérés. Le *CPL* est une technique puissante de traitement de la parole notamment utilisée pour l'encodage et la transmission [Rabiner 78]. Cette représentation a permis de créer une distance entre mots (basée sur la distance d'Itakura), distance finalement utilisée par l'algorithme *k-means* pour le clustering de mots.

Fu et al. [Fu 01] ont développé une représentation originale des séries temporelles, basée sur les Points Perceptuellement Importants (*PPI*). Un algorithme est proposé pour la recherche des *PPI*, ainsi qu'une distance entre ensembles de *PPI* représentant différentes séries temporelles. Le groupement des données pour la reconnaissance étant réalisé à l'aide de *self-organizing maps*, une classe de réseaux de neurones comprenant une représentation discrétisée de l'espace d'entraînement [Kohonen 97].

3.1.2 Approches de traitement de séries temporelles basée sur des modèles probabilistes

Les approches décrites dans ce paragraphe font l'hypothèse que les séries temporelles sont générées par des modèles ou par des mélanges de distributions de probabilité sous-jacents. Dans le premier cas, les méthodes ont recours principalement à des modèles auto-régressifs. Dans la suite de ce manuscrit, les travaux présentés utilisent principalement des modélisations décrivant les distributions sous-jacentes des processus et les approches développées sont donc différentes des modèles auto-régressifs (voir section 1.1). Nous nous concentrerons donc, dans la suite de cette section, sur les approches indiquant des modélisations sous-jacentes. Le lecteur intéressé par les modèles auto-régressifs pourra se référer à [Pandit 83, Box 08].

L'intérêt d'utiliser des modèles probabilistes pour le traitement de séries temporelles est la prise en compte de l'incertitude associée aux observations ainsi que la possibilité, notamment avec les MMC, de spécifier des modélisations adaptées aux différentes problématiques. Cette dernière remarque sera confirmée lors de la description des méthodes de traitement des trajectoires vidéos où l'utilisation de MMC dédiés (pour des tâches précises) est très fréquente.

Plusieurs approches caractérisent les séries temporelles à l'aide de distributions de probabilités. Un premier groupe d'approches décrit les séries temporelles à l'aide de mélanges de gaussiennes. L'algorithme utilisé pour l'estimation des paramètres de ces modélisations est l'algorithme *EM* (section 2.3)[Bilmes 98, Dempster 77] qui permet d'estimer la densité des individus à l'aide d'un modèle statistique choisi *a priori*. L'objectif de cet algorithme est d'estimer les paramètres d'un modèle de mélange de distributions de la forme :

$$f(x) = \sum_{i=1}^{K} c_i \mathcal{N}_i(x, \lambda_i),$$

où $\mathcal{N}$ est une gaussienne, c_i correspond au poids associé à $\mathcal{N}_i$ et λ_i est l'ensemble des paramètres de la distribution $\mathcal{N}_i$. Cette approche est utilisée pour la reconnaissance supervisée de séries temporelles, un modèle étant entraîné pour chaque classe de séries temporelles, la reconnaissance s'effectuant par maximum de vraisemblance [Povinelli 04].

Un certain nombre de méthodes ont également utilisé les MMC (chapitre 1) pour la reconnaissance et le clustering de séries temporelles. L'étude la plus approfondie concernant l'analyse de séries temporelles par MMC est l'œuvre de Bengio [Bengio 99]. Cette étude s'intéresse notamment à l'utilisation de différents types de MMC (dont les MMC continus "standards" et le MMC E/S, voir section 1.3.2) pour la reconnaissance de séries temporelles (et, notamment, la reconnaissance de la parole). Un modèle est, de manière similaire à l'utilisation de mélange de gaussiennes, entraîné pour chaque classe de séries temporelles et la reconnaissance (supervisée) se fait par maximum de vraisemblance.

De plus, plusieurs approches ont été proposées pour le clustering de séries temporelles. On pourra tout d'abord citer Li et Biswas [Li 00, Li 02] qui proposent un algorithme de clustering en quatre étapes dont les intérêts principaux sont tout d'abord de permettre aux MMC de changer dynamiquement de structure pendant le processus de clustering afin de mieux "coller" aux données et, d'autre part, de proposer un critère permettant de sélectionner automatiquement une partition des séries temporelles. Dans le même ordre d'idée, Smyth [Smyth 97] a défini une méthode originale pour l'initialisation de la matrice de transition A des MMC ainsi qu'une sélection automatique du nombre de clusters. Le lecteur intéressé par les MMC pour le clustering de séries temporelles pourra également se référer à [Alon 03], [Porikli 04b] et [Oates 01]. Il est à noter [Porikli 04a] que le clustering de séries temporelles par l'utilisation d'algorithme de type *EM* ou par MMC nécessite des séries temporelles de tailles importantes. En effet, ces deux modélisations "complexes" requièrent des ensembles de données d'entraînement importants, et ne peuvent effectuer de clustering efficace de trajectoires de tailles réduites (ce qui est souvent le cas dans des vidéos).

Dans cette thèse, il ne sera question que de données unidimensionnelles. Néanmoins, pour ceux intéressés par l'étude de séries temporelles multidimensionnelles, un travail de thèse a été conduit pour la modélisation de données multivariées [Kirshner 05] par des modèles de mélanges de gaussiennes ainsi que par des MMC.

3.2 Analyse de trajectoires vidéos

Nous nous intéressons aux techniques exploitant les trajectoires dans des séquences d'images ou vidéos, certaines méthodes présentées dans la section précédente consacrée aux séries temporelles (et introductive à cette section consacrée aux trajectoires vidéos) pourront être reconsidérées dans cette section. Néanmoins, nombre de méthodes "originales" et de moyens de représentation pour le traitement de trajectoires vidéos ont été proposés, en fonction principalement des objectifs applicatifs. Ainsi, nous avons décidé de décrire tout d'abord les moyens de caractériser les tra-

jectoires vidéos, avant de nous concentrer sur les modélisations (déterministes puis probabilistes, voir figure 3.5) qui en ont été faites. Le but n'est pas de faire une liste exhaustive des travaux traitant les trajectoires vidéo (bien qu'une large partie de ces travaux soit présentée dans cette section), mais plutôt de mettre en valeur les différentes directions proposées. De plus, certains travaux peuvent se classer dans plusieurs parties différentes.

3.2.1 Primitives de trajectoires vidéos considérées

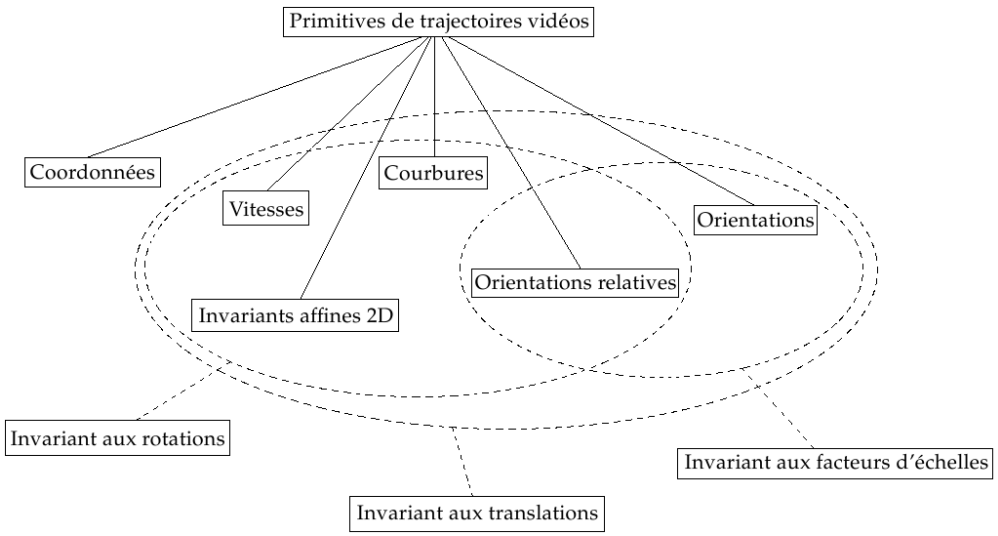

FIG. 3.4 – Présentation générale des primitives considérées pour le traitement de trajectoires vidéos.

De nombreux moyens de représentation de trajectoires dans des séquences d'images (ou vidéos) ont été proposés (Fig. 3.4), les premièrs (et les plus nombreuses) utilisant directement les coordonnées de ces trajectoires [Makris 05, Makris 02, Bashir 07a, Fu 05, Chang 98, Anjum 07a, Antonini 06, Piciarelli 05, Melo 04, Zelnicker 08, Zhou 08b, Nascimento 08]. De telles représentations permettent seulement de traiter les similarités spatiales "strictes" entre trajectoires et sont largement exploitées pour des tâches de vidéo-surveillance d'un site à l'aide d'une unique caméra fixe.

D'autres méthodes ont cherché à comparer des trajectoires 2D issues de vidéos en calculant les orientations locales [Porikli 04a, Li 06], les vitesses [Porikli 04a, Wang 06] ou les courbures [Junejo 07a, Lou 02]. Néanmoins, la représentation à l'aide d'orientations locales ne possède pas la propriété d'invariance à la rotation dans le plan image. De même, les vitesses observées, tout comme les courbures, dépendent de la distance à la caméra de l'action filmée. Les auteurs de [Chen 08] ont considéré la représen-

tation “null space representation”, une représentation permettant de définir des invariants aux transformations affines 2D (donc aux transformation de rotation et de translation). Néanmoins, cette représentation ne permet pas une invariance au facteur d’échelle. Certaines représentations intéressantes exploitent, en plus de certaines de ces primitives, des informations de densité de couleur (information de type *RGB*) [Hu 06, Hu 07] ou la taille des objets suivis [Izo 07a, Izo 07b].

Toutes ces méthodes possèdent des caractérisations qui ne sont pas invariantes simultanément aux transformations de translation, de rotation et de changement d’échelle (dans le plan image). Un premier travail de Bashir et al. [Bashir 06] propose deux représentations invariantes à la translation et à l’orientation, dans lesquelles les effets d’échelle peuvent être gérés en rééchantillonnant les trajectoires à une taille commune. Fashandi et al. [Fashandi 05] ont eux adoptée une représentation considérant la séquence relative des angles. Cette dernière représentation permet d’obtenir une représentation “directe” (sans besoin de rééchantillonnage) invariante aux transformations de translation, de rotation et d’échelle dans le plan image, ce qui peut s’avérer (comme on le verra dans la partie II) d’un intérêt primordial pour certaines applications en analyse de vidéos.

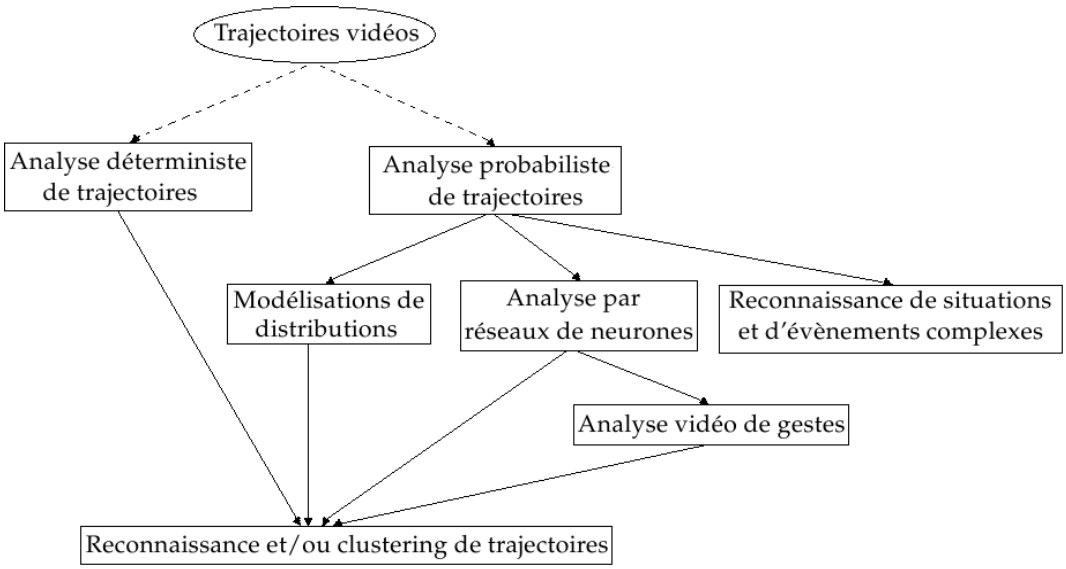

FIG. 3.5 – Illustration générale des méthodes de traitement des trajectoires vidéos décrites dans les sections 3.2.2 et 3.2.3.

3.2.2 Modélisations déterministes de trajectoires vidéos

Cette section présente les modélisations déterministes de trajectoires vidéos. Nous avons décidé, pour plus de clarté, de diviser les méthodes existantes en deux sous-parties correspondant tout d'abord aux méthodes pour la reconnaissance supervisée de trajectoires, puis aux techniques dédiées au clustering de trajectoires vidéo.

3.2.2.1 Reconnaissance supervisée déterministe de trajectoires vidéos

Shim et al. [Shim 04] ont développé un logiciel permettant de trouver les instances de trajectoires vidéo similaires à celle proposée par un utilisateur, dont la méthode est basée sur une représentation des directions du mouvement et des distances parcourues ainsi que sur une distance "k-warping" inspirée par les méthodes de classification de séries temporelles de type *DTW* (section 3.1.1). Cette méthode se compare avec succès à des méthodes très proches de traitement de trajectoires [Li 97, Shan 98].

Une méthode de reconnaissance de trajectoires reposant sur une segmentation des trajectoires en sous-trajectoires a été proposée dans [Chen 00, Chang 98]. Un algorithme de reconnaissance des séquences de ces sous-trajectoires décrit dans [Smith 99] est utilisé. Chen et al. [Chen 04] ont également défini une méthode d'indexation de trajectoires utilisant une représentation des trajectoires par sous-trajectoires en considérant des informations de direction et de quantité de mouvement. Ces sous-trajectoires sont ensuite représentées à l'aide de vecteurs fréquentiels (voir [Kahveci 01] pour la définition de ces vecteurs fréquentiels), une distance entre vecteurs fréquentiels étant alors développée et exploitée pour la reconnaissance de trajectoires.

Bashir et al. ont développé une méthode utilisant un pré-traitement permettant l'invariance aux translations, rotations et au cisaillement [Bashir 04]. Une représentation des trajectoires par des descripteurs issus d'une transformée de Fourier est utilisée dans une première phase de reconnaissance (distance euclidienne sur les coefficients de Fourier) permettant de choisir un faible nombre de proches voisins. Les trajectoires sont alors découpées en sous-trajectoires à l'aide des courbures, avant un traitement en analyse par composantes principales (ACP) [Bashir 07a] et la reconnaissance finale parmi les proches voisins issus de la première étape.

Une méthode déterministe a été proposée pour la surveillance de sites à l'aide d'une caméra [Makris 02] dans laquelle les trajectoires observées sont géométriquement comparées à des routes correspondant aux chemins de passage. Ces routes sont créées en groupant les trajectoires géométriquement similaires et mises à jour en fonction des trajectoires observées qui leur sont attribuées, de nouvelles routes étant créées lorsque les trajectoires observées ne sont pas assignées à une route connue. La reconnaissance d'une trajectoire à une route étant faite par calcul de distances entre la tra-

jectoire et l'enveloppe de la route.

Moënne-Locoz et al. [Moënne-Loccoz 06] ont utilisé des Séparateurs à Vaste Marge (SVM, [Burges 98]) pour la reconnaissance de vidéo. Les SVM sont ici appliqués à des histogrammes multi-échelles d'ensembles de trajectoires, chacun de ces ensembles de trajectoires étant représentatif d'une vidéo donnée.

3.2.2.2 Clustering déterministe de trajectoires vidéos

Vlachos et al. [Vlachos 02, Vlachos 02b, Buzan 04] ont eu recours à la distance *LCSS* (section 3.1.1) et à une représentation invariante à la translation pour du clustering hiérarchique. Une technique reposant sur les histogrammes des orientations a été proposée dans [Li 06] ainsi qu'un clustering hiérarchique basé sur la distance de Bhattacharyya.

Junejo et al. [Junejo 04, Junejo 07a] ont exploité un clustering initial hiérarchique utilisant la distance de Hausdorff entre trajectoires. Une méthode de classification est ensuite mise en place pour la détection d'évènements anormaux en terme de localisation, de forme et de dynamique des trajectoires. Melo et al. [Melo 04] ont exploité, eux aussi, la distance de Hausdorff et un clustering par algorithme *k-means*(voir section 2.4).

Izo et al. [Izo 07a, Izo 07b] ont fait appel à un clustering spectral basé sur une représentation à descripteurs multiples (tels que la taille de l'objet suivi, la position, la vitesse...) pour des tâches de vidéo-surveillance. Dans le même domaine d'applications, les auteurs de [Piciarelli 05] ont utilisé une distance euclidienne modifiée pour un clustering en temps réel ainsi que la détection d'évènements inattendus [Piciarelli 06]. Une structure d'arbre permet, au fur et à mesure du temps, de regrouper les clusters initiaux pour modéliser les chemins possibles, et ce avec une caméra fixe. Piciarelli et al. ont également mis en place des méthodes de clustering [Piciarelli 07] et de détection anormaux [Piciarelli 08] en utilisant les Séparateurs à Vaste Marge (SVM, [Burges 98]) et plus spécifiquement les ν-SVM, les coordonnées de chaque trajectoire étant rééchantillonnées afin d'être décrites par un vecteur de taille commune.

L'algorithme *mean-shift* a été utilisé par Anjum et al. [Anjum 07a, Anjum 08] pour le clustering de vidéos, les coordonnées des trajectoires de véhicules ayant été reconstruites dans le plan au sol. Une technique permettant de modéliser des scènes de trafic de véhicules et de piétons a été développée par Wang et al. [Wang 06]. Une première mesure de similarité est considérée, spécialement conçue pour distinguer piétons et véhicules, puis les observations de chacun des deux groupes obtenus sont ensuite soumises à une seconde phase de clustering en exploitant une seconde mesure de similarité prenant en compte tout autant les coordonnées observées que les tailles d'ob-

jets et les vitesses. Ce clustering initial est ensuite utilisé afin de trouver les directions empruntées et ainsi réaliser une classification en temps réel.

Toutes ces méthodes déterministes ont été uniquement proposées pour des tâches de vidéo surveillance avec caméra unique et fixe ainsi que pour des tâches de recherche de trajectoires dans des bases de données. En effet, ces méthodes correspondent à des comparaisons "exactes" de trajectoires, par le biais des ressemblances géométriques entre trajectoires (ou représentations de trajectoires), et nécessitent, en pratique, des trajectoires de tailles sensiblement égales. De plus, elles ne permettent pas de modéliser les causalités inhérentes aux comportements associés aux trajectoires ainsi qu'une prise en compte de l'incertitude associée aux observations que peuvent fournir les modélisations probabilistes telles que celles décrites dans la suite.

3.2.3 Modélisations probabilistes de trajectoires vidéos

Cette section présente les modélisations non-déterministes (ou probabilistes, aléatoires, stochastiques...) de trajectoires vidéos. Nous avons décidé, une fois encore pour plus de clarté, de diviser les méthodes existantes en deux sous-parties. Dans cette section, la première partie sera consacrée aux méthodes pour la reconnaissance supervisée et non-supervisée de trajectoires avec des modélisations aléatoires. La seconde partie s'intéressera aux reconnaissances supervisée et non-supervisée (ou clustering) de trajectoires utilisant des réseaux de neurones.

Le clustering de trajectoires vidéos, avec de telles méthodes probabilistes, s'avère complexe. La principale raison est que les individus (les trajectoires vidéo) à grouper sont souvent composées de seulement quelques dizaines d'observations (à la différence des séries temporelles qui sont en général formées par un nombre conséquent d'observations). Alors, les algorithmes généralement utilisés pour l'entraînement des paramètres de ces modèles probabilistes nécessitent souvent un nombre important d'observations afin de fournir une modélisation adaptée des trajectoires.

3.2.3.1 Reconnaissance de trajectoires vidéos à l'aide de modélisations statistiques

Plusieurs méthodes ont été proposées pour la modélisation probabilistes de chemins dans des vidéos acquises par des caméras fixes en vidéo-surveillance.

Une méthode a ainsi été développée par Stauffer et Grimson [Stauffer 00] qui ont appliqué un mélange de gaussiennes en chaque pixel afin de modéliser son "histoire". L"'histoire" à un temps t associée à un pixel correspond aux couleurs observées en ce pixel dans la séquence d'image, pour $t = 0 \dots t - 1$. Une soustraction du fond est réalisée, un dictionnaire représentant les différents chemins empruntés est créé et utilisé pour le clustering, à l'aide d'une classification hiérarchique (voir section 2.2) par

co-occurences, des données observées.

Une méthode récente et intéressante a été proposée par Wang et al. [Wang 08]. Ils ont considéré une approche bayésienne non-paramétrique incluant une extension des processus de Dirichlet hiérarchiques (appelée processus de Dirichlet hiérarchique dual) pour permettre le clustering de données. Ces processus de Dirichlet définissent des mots trouvés dans des documents en générant un dictionnaire, et ont été utilisés avec succès en traitement de la parole. Les trajectoires (les documents) sont divisées en régions sémantiques (les mots), l'extension faite dans ce travail permettant ensuite le clustering de documents et, donc, de trajectoires.

Les mélanges de gaussiennes ont été plus spécifiquement utilisés dans différents travaux pour modéliser les chemins de passage (avec une seule caméra) et, ainsi, pouvoir effectuer une vidéo-surveillance automatique d'une place, d'un parking. Hu et al. [Hu 06] exploitent eux aussi, en plus des coordonnées observées, les vitesses et les tailles des objets suivis. Ils proposent également un clustering en deux phases, une première phase n'utilisant que les informations de coordonnées, puis une seconde prenant en compte les informations temporelles et, ainsi, les dynamiques (en terme d'évolutions de la vitesse) observées à l'aide d'un algorithme de type *k-means*. Ce clustering est ensuite utilisé pour l'estimation d'un mélange de gaussiennes en vue de la classification de trajectoires et la détection d'anomalies. Ainsi, cette méthode propose un clustering déterministe (et, donc, soumis aux limitations déjà soulignée en section 3.2.2) utilisé, par la suite, pour une classification probabiliste des trajectoires.

Jung et al. ont étendu l'utilisation des mélanges de gaussiennes au clustering de trajectoires à l'aide des déplacement relatifs observés. Ils ont également mis en place une méthode de détection d'évènements inhabituels, par l'étude des coordonnées et de la vitesse des trajectoires, à l'aide d'histogrammes [Jung 08]. Les auteurs de [Prati 08] ont eu recours aux distributions de von Mises et les statistiques circulaires qui permettent de représenter les distributions des directions des trajectoires considérées. L'algorithme *EM* est ensuite étendu à l'estimation des paramètres de mélanges de distribution de von Mises pour la reconnaissance de trajectoires par maximum *a posteriori*. Le clustering réalisé dans cette méthode ne permet pas de prendre en compte les causalités temporelles devant permettre de caractériser les trajectoires.

3.2.3.2 Reconnaissance supervisée et non-supervisée probabiliste de trajectoires vidéo à l'aide de réseaux de neurones

Les méthodes décrites dans cette section correspondent à une sous-ensemble des méthodes de reconnaissance de trajectoires vidéos à l'aide de modélisations statistiques (voir section précédente). En effet, les méthodes d'apprentissage par réseaux de neurones sont des méthodes statistiques.

Khalid et Naftel [Naftel 06a, Naftel 06b, Khalid 05] ont comparé des représentations des trajectoires dans différents espaces (espace des polynômes, représentation de Chebyshev, transformée de Fourier discrète) qui sont ensuite clusterisées à l'aide de cartes auto-organisatrices [Kohonen 97].

Selon une idée assez similaire à celle de Stauffer et al. [Stauffer 00], Johnson et Hogg [Johnson 95] ont développé une méthode exploitant les coordonnées et les vitesses des objets suivis. Elle permet de produire des clusters de trajectoires à l'aide de réseaux de neurones [Kohonen 97] et d'extraire des "prototypes" représentant les principaux types de trajectoires observées. Ces prototypes sont ensuite utilisés pour la reconnaissance en temps réel des événements observés. Le même type d'approche [Bauer 06] a été proposé, en considérant les positions, les vitesses et les directions des trajectoires observées, utilisant cette fois des "Growing Neural Gas" [Fritzke 95] ainsi qu'une carte hebbienne (représentant la proximité entre prototypes, [Fritzke 95]) pour la production de prototypes.

Les réseaux de neurones ont été utilisés par Hu et al. [Hu 04a, Hu 04b] pour la reconnaissance d'événements, la détection d'événements inattendus et la prédiction de mouvements. Les réseaux de neurones à délai temporel (ou "Time Delay Neural Networks", TDNN, voir [Waibel 89]) ont également été appliqués à la reconnaissance, à l'aide des trajectoires vidéos des mains, de gestes manuels [Yang 02].

Les réseaux de neurones sont souvent utilisés comme des "boites noires", peu de contraintes étant imposées sur les topologies de telles modélisations. Les MMC sont plus particulièrement exploités afin de contrôler de façon précise la modélisation de trajectoire produite en fonction de la connaissance *a priori* sur les données observées. Ceci est réalisé à l'aide d'architectures (ou topologies) particulières permettant de faciliter l'apprentissage de tels modèles (voir chapitre 1).

3.2.3.3 Reconnaissance supervisée et non-supervisée probabiliste de trajectoires vidéo à l'aide de MMC

Les méthodes utilisant des MMC appartiennent à la classe des réseaux de neurones, les méthodes rencontrées dans cette section correspondent donc à un sous-ensemble de la section précédente. Ainsi, des méthodes pour la vidéo surveillance à l'aide de MMC continus (utilisés pour la reconnaissance d'événements par maximum de vraisemblance) ont été proposées [Brand 00, Nascimento 07, Remagnino 01, Dockstader 06, Aleotti 05]. Nous présentons dans cette section les méthodes qui nous paraissent les plus intéressantes.

Bashir et al. [Bashir 07b] segmentent tout d'abord les trajectoires à partir des courbure, les sous-trajectoires obtenues étant ensuite caractérisées par les coefficients obtenus à l'aide d'une analyse en composantes principales. Ces coefficients sont ensuite représentés, pour chaque classe de trajectoires, à l'aide de MMC continus (et également, pour des raisons de comparaison à ces MMC, par des mélanges de gaussiennes), la reconnaissance de trajectoires se faisant par maximum de vraisemblance. Une version de cette méthode incluant un pré-traitement des trajectoires pour l'invariance à la rotation, à la translation et au cisaillement est également proposée [Bashir 06].

Lou et al. [Lou 02] ont abordé la reconnaissance d'activité dans des vidéos au moyen de la distance d'Hausdorff entre trajectoires et une méthode de clustering hiérarchique en deux phases, une première considérant les informations spatiales et une seconde les informations de dynamique. Une méthode pour la segmentation de trajectoires en phases sémantiques est également mise en place, celle-ci étant effectuée à l'aide de MMC continus (une méthode assez identique peut être trouvée dans [Fraile 98]).

Chan et al. ont développé deux modélisations pour la reconnaissance et la détection d'événements dans des aéroports [Chan 04, Chan 06a]. La première utilise des MMC modélisant les proximités entre éléments observés (mobiles ou fixes). La seconde exploite des réseaux de neurones dynamiques [Murphy 02] pour simultanément reconstruire les parties manquantes de trajectoires [Chan 06b] (en cas d'occlusions ou de suivi intermittent) et la reconnaissance d'événements [Chan 06a].

Des travaux d'intérêt ont été développé par Fatih Porikli [Porikli 04a] dans lesquels il a exploité les MMC pour la modélisation de trajectoires. Des représentations variées (avec propriété d'invariances) ainsi qu'une distance entre trajectoires ont été mise en œuvre. Cette distance a été appliquée au clustering [Porikli 04b] et à la détection d'événements rares [Porikli 04c].

Enfin, Berclaz et al. [Berclaz 08] proposent une méthode "contextuelle" d'analyse des trajectoires par MMC dans une scène à l'aide de plusieurs vidéos. En effet, après avoir discrétisé la scène observée en zones, des cartes de mouvements sont définies pour chaque comportement. Un état caché correspond, dans cette méthode, à une position donnée associée à un comportement particulier, et chacune des cartes correspond aux transitions entre zones. Ensuite, ce modèle est utilisé pour la reconnaissance de comportements, par maximum de vraisemblance, et la détection de trajectoires inattendues, à l'aide de seuil de vraisemblance.

Les MMC ont donc été largement utilisés pour la modélisation et la reconnaissance de trajectoires dans des vidéos. Ils offrent en effet des algorithmes d'apprentissage et

de reconnaissance efficaces. Ils permettent également, au contraire des réseaux de neurones, de définir une architecture aidant à la prise en compte d'informations *a priori* ainsi qu'à l'entraînement des modèles.

∗ *Reconnaissance de gestes dans des vidéos à l'aide de MMC*

Les méthodes d'apprentissage par MMC ont également été appliquées à la reconnaissance de gestes dans des vidéos à l'aide de trajectoires [Starner 95, Min 04]. Nous présentons rapidement quelques uns de ces travaux.

Meyer et al. [Meyer 98] ont notamment considéré les vitesses de quatre parties du corps (la tête, le cou, la hanches et le genou) ainsi qu'une modélisation par MMC continus pour la classification de démarches. Les auteurs de [McKenna 03] et [Psarrou 02] ont également exploité les MMC continus pour la modélisation de gestes à partir des trajectoires de mains. Les résultats obtenus ont été favorablement comparés à ceux d'autres méthodes (proches de celle proposée pour la reconnaissance de gestes humains dans [Black 98]). Wilson et Bobick [Wilson 99] ont défini une nouvelle modélisation par MMC pour la reconnaissance de gestes, les MMC paramétriques (MMCP) dont l'avantage principal est d'introduire un paramètre (associé aux probabilités d'observation des MMC) permettant, par exemple, de modéliser "finement" les variations d'amplitude dans des gestes (gestes du type "grand comme ça"). Vogler et Metaxas [Vogler01] ont appliqué les MMC Pa (section 1.3.4) à des données issues de suivi 3D pour la reconnaissance de mots dans le langage des signes. Ils analysent les mouvements des mains et exploitent des notions d'ouverture et de fermeture des mains (information importante dans la compréhension du langage des signes). Enfin, Just et al. [Just 04] ont développé une méthode mettant en œuvre des MMC E/S (voir section 1.3.2) pour la reconnaissance de gestes à l'aide des trajectoires 3D reconstruites (à l'aide de plusieurs caméras) de la tête, du torse et des mains.

3.2.4 Reconnaissance de situations et d'événements "complexes" à partir des trajectoires

3.2.4.1 Introduction

La partie III de ce document porte sur l'analyse d'informations "complexes" dans des vidéos, et plus particulièrement sur la segmentation de vidéos de sport en phases d'intérêt. Ainsi, nous proposons dans cette section un état de l'art des méthodes existantes pour la compréhension d'évènements "complexes" dans des vidéos.

Des travaux récents exploitent donc les informations visuelles afin d'effectuer une analyse de haut niveau des situations observées. Des techniques issues de l'intelligence artificielle sont associées à des méthodes de vision par ordinateur pour la compréhension de séquences d'événements dans des vidéos. Les trajectoires vidéos sont le plus souvent exploitées pour la compréhension de scènes sémantiquement complexes.

3.2.4.2 Description des méthodes

Remagnino et al. [Remagnino 98] ont eu recours aux réseaux bayésiens pour la surveillance de parking. Dans ce travail, les auteurs ont modélisé les comportements d'objets en mouvement (appelés agents) à partir de leurs trajectoires. Les réseaux bayésiens permettent de prendre en compte les données de coordonnées, de vitesse, d'accélération et de forme des mouvements des agents. De plus, ces réseaux bayésiens permettent de modéliser les interactions entre agents et définissent si les agents ont des mouvements similaires ou différents (opposés ou non). Liu et al. [Liu 06] ont également considéré une architecture multi-agent pour la reconnaissance d'activités à l'aide de trajectoires dans des vidéos. Ils proposent une modélisation à l'aide de MMC, appelé "MMC à observations décomposées", dont le principe est que chaque état génère des observations indépendantes pour chacun des agents, et proposent une extension de l'algorithme EM pour l'estimation des paramètres de ce modèle. L'interaction entre au moins deux agents a été développée par Oliver et Pentland [Oliver 00] pour la modélisation d'événements vidéos. Les MMC couplés ont permis de prendre en compte les interactions observées entre agents à partir de leurs trajectoires.

Des modélisations statistiques définissant des scénarios *a priori* ont également été mise en œuvre pour la reconnaissance d'actions complexes à l'aide de trajectoires dans des vidéos par Hongeng et al. [Hongeng 01, Hongeng 03a]. La représentation des événements se fait par quatre couches hiérarchiques, la première correspondant au traitement des images pour la détection et le suivi d'objets (ou agents), la seconde à la mise à jour de propriétés associées aux agents (distance entre objets, évolution des vitesses et des directions des objets observés...). Les troisième et quatrième couches définissent les scénarios, la première pour les événements simples (approcher un autre objet, ralentir). La dernière utilise ces événements simples pour la modélisation d'actions complexes en séquences d'événements simples et pour la modélisation de séquence d'actions complexes, ce à l'aide de règles logiques d'enchaînements et de durées. Un travail intéressant a utilisé ce type de hiérarchie pour la modélisation et la détection d'événements de durées variables à l'aide de modélisations semi-markoviennes des trajectoires vidéos [Hongeng 03b].

Certaines méthodes proposées pour la modélisation et la reconnaissance d'activités définissent un certain nombre de comportements possibles *a priori* entre deux objets mobiles et, donc, entre deux trajectoires. Tout d'abord, Niu et al. [Niu 04] a défini trois comportements ("suivre une personne", "rejoindre et continuer avec une personne" et "harcèlement"), de façon très simple, à l'aide des positions et des vitesses relatives entre les deux personnes. Zhou et al. ont récemment [Zhou 08] repris ces travaux en utilisant les positions, la courbure et la vitesses des trajectoires observées, et en définissant cinq comportements ("la poursuite", "suivre une personne", "mouvement indépendant", "la rencontre" et le "déplacement identique"). La reconnaissance

de ces comportements est obtenue par des tests de causalité de Granger [Granger 69], tests utilisés généralement en statistique économique.

D'autres travaux tentent de décrire de façon générale (pour le traitement haut niveau d'informations visuelles) les lois de compréhension des événements à l'aide de grammaires non-contextuelles [Minnen 03, Zhang 08]. Nevatia et al. [Nevatia 04, Natarajan 05, François 05] notamment ont mis en place de telles grammaires pour l'annotation d'événements vidéos. Les événements sont marqués par un (ou des) changement(s) dans les vidéos traitées. Un langage de représentation des événements vidéo (Video Event Representation Language, VERL) a été défini. Celui-ci est basé sur la mise en place, à partir d'événements élémentaires combinés et d'opérations simples (par exemple des opérations d' "alternance", de "séquençage"ou de "répétition"...), d'événements complexes. Une telle ontologie a également été exploitée dans la description de scénarios (prédéfinis) par Vu et al. [Vu 02], avec applications à la surveillance de métro.

Des méthodes statistiques ont aussi été considérées pour la description "grammaticale" d'événements visuels à l'aide de trajectoires vidéo, appliquées à la surveillance de parking [Ivanov 00] ou de scène de jeux [Moore 01]. La reconnaissance d'événements est effectuée à l'aide d'une modélisation en deux phases. Une première de bas niveau correspond à la détection d'événements simples. La seconde phase incorpore la grammaire considérée, permettant de mettre en œuvre des contraintes temporelles (pour lever certaines ambiguïtés sur les événements observés au bas-niveau) et d'introduire de la connaissance *a priori* sur la structure d'enchaînement et de transition entre événements élémentaires. Un travail traitant les trajectoires vidéos pour la reconnaissance d'événements en vidéo surveillance, et utilisant des règles de grammaire entre évènements simples à l'aide du critère *MDL* ("minimum description length") a également été proposé par Zhang et al. [Zhang 07].

Un travail intéressant a été développé par Tran et al. [Tran 08], utilisant les "Réseaux Logiques Markovien" (RLM, voir [Richardson 06]). Ces RLM permettent de créer un lien entre les grammaires logiques provenant principalement du domaine de l'intelligence artificielle (et dont on a décrit les utilisations faites en reconnaissance de situations dans la présente section) et les modélisations probabilistes souvent exploitées en vision par ordinateur. En effet, dans ces réseaux, chaque formule logique est associée à une probabilité d'observation modélisant l'incertitude de l'analyse effectuée sur les événements, cela permettant de traiter plus efficacement des données bruitées ou manquantes [Tran 08].

Définir des hiérarchies *a priori* peut aider à modéliser des évènements, *a fortiori* lorsque les événements que l'on désire modéliser peuvent être qualifiés de complexes [Nevatia 03]. Shi et al. ont utilisé des réseaux de propagation [Shi 04] pour le traitement de trajectoires à l'aide de scénarios. Certaines modélisations ont également été développées à cet effet, telles que celles définies par Natarajan et al. [Natarajan 07a] et [Natarajan 07a] pour le traitement de trajectoires vidéos. La première, appliquée à la reconnaissance de langage des signes, introduit une hiérarchie à l'aide de modèles semi-markoviens et une couche inférieure composée de MMC pour la modélisation d'interactions dans des activités entre humains. La seconde expose une modélisation couplée de modèles semi-markoviens pour la reconnaissance, là encore, de Language des signes. Les auteurs de [Robertson 05] ont développé une modélisation à l'aide de MMC pour la reconnaissance d'actions dans des vidéos de tennis par analyse de trajectoires ainsi qu'une description visuelle des acteurs. La structure du MMC intègre les règles du tennis afin de mieux modéliser les transitions entre actions.

Les notions de hiérarchies et de scénarios sont donc généralement mises en œuvre pour la compréhension de situations et d'évènements complexes dans des vidéos. En effet, de telles notions permettent aux modèles développés d'être guidés dans l'apprentissage et la compréhension de telles situations. Ainsi, dans la partie III de ce document, nous proposerons des modèles markoviens hiérarchiques permettant de prendre en compte les phases de jeu *a priori* connues afin de comprendre les scénarios sportifs observés.

3.3 Conclusion

Ce chapitre montre l'intérêt grandissant de l'étude des séries temporelles et plus particulièrement des trajectoires issues de vidéos, les trajectoires vidéos correspondant à une information de relativement haut niveau. Celles-ci peuvent être utilisées tout autant pour des tâches d'extraction de "chemins" en vidéo-surveillance, de reconnaissance de contenu de vidéos dans des archives audiovisuelles, et même, pour les travaux les plus récents, de reconnaissance de situations complexes et de scénarios. Pour diverses raisons, notamment de prise en compte de l'incertitude attachée aux données et d'efficacité de mise en œuvre, les MMC ont été largement privilégiés. Ils permettent aussi de définir des modélisations dédiées (selon les types de MMC) aux problématiques envisagées. Ainsi, dans la suite de nos travaux, nous avons privilégié les MMC pour le traitement des trajectoires vidéos.

Conclusion

Nous avons présenté, dans ce premier chapitre, un état de l'art des principaux modèles de Markov utilisés pour la modélisation des causalités temporelles en vision par ordinateur. En effet, notre volonté étant d'utiliser les trajectoires observées dans des vidéos pour la reconnaissance de contenus dans des vidéos, nous serons amenés à prendre en compte les causalités temporelles inhérentes aux trajectoires. La variété des modélisations markoviennes doit permettre de traiter les dépendances dans un voisinage temporel et, donc, les causalités temporelles des trajectoires.

Dans le chapitre 2, nous avons fourni un court état de l'art des méthodes de classification non-supervisée de données, tâche qui sera considérée ultérieurement dans nos travaux.

L'importance des modélisations markoviennes, décrites dans le chapitre 1, est mise en valeur dans le chapitre 3, qui présente un état de l'art des méthodes existantes pour le traitement des séries temporelles dans des vidéos à des fins de reconnaissance de contenus. En effet, l'intérêt des MMC est démontré par leur utilisation dans un grand nombre de travaux, afin de représenter les dépendances temporelles. Ainsi, dans la suite de ce document, les modélisations markoviennes seront largement exploitées.

Le chapitre 3 a également permis de spécifier le cadre de travail, c'est-à-dire l'analyse des trajectoires observées dans des vidéos. La mise en place du contexte de notre travail a également permis de souligner les limitations et les difficultés rencontrées lors de tels travaux, et que nous pourrons tenter de résoudre dans la suite de ce manuscrit.

La partie suivante est consacrée à la représentation et à la modélisation de trajectoires extraites de plans vidéos.

Deuxième partie

Reconnaissance d'évènements vidéos à l'aide de trajectoires

Introduction

Dans cette partie, nous décrivons la méthode originale d'analyse de trajectoires extraites de vidéos pouvant être issues de caméras mobiles. En effet, de nombreux contenus vidéos sont filmés par des caméras en mouvement. De plus, des contenus identiques peuvent être recherchés dans des vidéos correspondant à différents angles de vue ou positions de caméra. Des biais en termes de translation, d'échelle et de rotation dans le plan vidéo sont généralement observés suite à la compensation du mouvement de la caméra introduite dans l'extraction des trajectoires. Ainsi, nous avons élaboré une représentation adaptée des trajectoires, incluant un certain nombre d'invariances.

De plus, les trajectoires d'objets mobiles observées dans des plans vidéos sont souvent de longueurs assez courtes. La difficulté est alors d'avoir une modélisation suffisamment flexible, pouvant traiter de petites trajectoires tout en prenant en compte leurs causalités temporelles.

Cette seconde partie est organisée de la manière suivante :

- le **chapitre 4** présente tout d'abord succinctement les méthodes utilisées pour l'extraction des trajectoires. Ensuite, nous décrivons la représentation de trajectoire que nous avons adopté. Elle explicite les propriétés de dynamique et de forme des trajectoires vidéos. La sélection des paramètres impliqués est également considérée ;

- dans le **chapitre 5**, nous proposons une modélisation probabiliste, inspirée des modèles de Markov cachés, permettant d'appréhender un nombre réduit de données. Une distance entre trajectoires est adoptée pour les tâches de reconnaissance de contenu vidéo ;

- les méthodes proposées dans les chapitres 4 et 5 ont été utilisées, dans le **chapitre 6**, pour des tâches de classification, de clustering, ainsi que de détection d'évènements inattendus à partir des trajectoires extraites de plans de vidéos de sport. Des tâches de classification et de clustering de formes dans des images ont également été abordées.

Chapitre 4

Définition des primitives caractérisant les trajectoires

"Ce qui pouvait être vrai, que le niveau humain fût plus élevé aprés qu'avant la révolution. La seule preuve du contraire était la protestation silencieuse que l'on sentait dans la moelle de ses os, c'était le sentiment instinctif que les conditions dans lesquelles on vivait étaient intolérables et, qu'à une époque quelconque, elles devaient avoir été différentes."

George Orwell - *1984*

Ce chapitre est dédié à la caractérisation des trajectoires considérées dans la suite de cette partie. Contrairement à la plupart des représentations proposées pour la caractérisation de trajectoires (voir 3.2.1), nous visons ici une représentation invariante à certaines transformations pertinentes dans le cadre du traitement de vidéos. De plus, nous désirons une représentation permettant de dégager des propriétés relatives aux dynamiques (en termes d'évolution de la vitesse au cours du temps) et aux formes des trajectoires traitées. Enfin, nous voulons élaborer une méthode d'extraction automatique des primitives (avec fixation des paramètres impliqués) qui puisse prendre en compte le bruit contenu dans les séquences observées. Ainsi, dans ce chapitre, nous décrirons tout d'abord les types de trajectoires qui seront considérés dans cette partie de la thèse. Ensuite sera présentée la représentation choisie (et donc, les primitives), ses propriétés ainsi que son mode de calcul. Enfin, cette analyse sera complétée par un ensemble de résultats permettant d'appréhender les propriétés des primitives retenues, dans un cadre réaliste. On examinera également les problèmes relatifs à l'extraction de ces primitives.

4.1 Suivi temporel et trajectoires vidéos

Le but de cette section est de présenter brièvement les procédures de vision par ordinateur utilisées sur des plans vidéos, filmées par des caméras mobiles, permettant d'extraire les trajectoires qui seront être utilisées dans la suite de ce manuscrit pour des tâches de reconnaissance de contenu.

4.1.1 Méthodes de suivi et d'estimation du mouvement de la caméra utilisées

Nous décrivons brièvement la technique de compensation robuste du mouvement de fond ainsi que la méthode de suivi par filtrage particulaire utilisées pour extraire les trajectoires des vidéos filmées par des caméra mobiles, dans les exemples de Formule 1 et de ski.

Il est important de préciser que les méthodes présentées pour le suivi et l'estimation du mouvement de la caméra ont été développées par des membres de l'équipe et intégrées dans un logiciel. Elles ne correspondent pas à nos travaux. Ainsi, de plus amples détails pour l'estimation robuste du mouvement dominant (et également pour la prise en compte des incertitudes pouvant être rencontrées lors de cette procédure) ainsi que pour l'algorithme de suivi par filtrage particulaire pourront être trouvés dans [Gengembre 08].

⋆ *Estimation robuste du mouvement de la caméra*

Le mouvement du fond est pris en compte et compensé dans la procédure de suivi décrite ci-dessous, les trajectoires étant ainsi recalées dans le plan image initial. Nous présentons ici les éléments principaux de la méthode utilisée.

Les données d'entrée utilisées pour l'estimation du mouvement de la caméra sont les déplacements estimés pour des points d'intérêt. Le calcul des vecteurs de déplacement des points d'intérêt est donné par une procédure KLT pyramidale [Bouguet 99]. Les points d'intérêt retenus sont alors ceux pour lesquels la confiance dans les vecteurs de déplacement calculés est la plus grande [Shi 94].

Les vecteurs de déplacement associés aux N points d'intérêt retenus sont notés $\mathbf{w}_i = (u_i, v_i)^T, i = 1...N$. Un modèle de mouvement à 3 paramètres est considéré, qu'englobe la translation horizontale (paramètre θ_1), le zoom (paramètre θ_2) et la translation verticale (paramètre θ_3). L'estimation des modèles paramétriques de mouvements de caméra a fait l'objet de nombreux travaux [Grinias 02, Okuma 04, Joly 96], et notamment par Odobez et al. [Odobez 94, Odobez 95] qui ont proposé une méthode d'estimation de modèles paramétrés de mouvement associant un estimateur robuste, du type M-estimateur, avec un schéma multi-résolution. Le modèle global affine du

mouvement, proche de celui de la méthode proposée par Okuma et al., est donné par :

$$\mathbf{w}_i = \begin{pmatrix} u_i \\ v_i \end{pmatrix} = X_i\,\theta = \begin{pmatrix} 1 & x_i & 0 \\ 0 & y_i & 1 \end{pmatrix}\theta,$$

avec $\theta = (\theta_1, \theta_2, \theta_3)$. L'hypothèse faite est que la plus grande partie des points d'intérêt se trouve sur le fond et ont des déplacements dus au mouvement de la caméra. Une méthode d'estimation robuste, basée sur un M-estimateur, mise en œuvre par une technique de moindres carrés pondérés itérés [Huber 81], permet d'éliminer l'influence des mouvements secondaires dus à des objets en mouvement dans la scène.

⋆ *Algorithme de suivi par filtrage particulaire*

La méthode de suivi utilisée pour l'extraction des trajectoires correspond à une méthode de filtrage particulaire permettant, entre autres, de gérer les situations suivantes : changements d'apparence de la cible, présence de fouillis, changements d'illumination, présence d'occultations, prise en compte des mouvements de la caméra et suivi d'objets multiples. Nous décrivons succinctement la version utilisée pour le suivi mono-objet avec prise en compte des mouvements de la caméra, une extension au suivi multi-objets pouvant être trouvée dans [Gengembre 08].

L'initialisation de l'algorithme de suivi est effectuée à la main, en traçant un rectangle (ou une ellipse) dans une image initiale autour de l'objet que l'on désire suivre. Le but est donc d'évaluer sa position, à chaque pas de temps et à l'aide des propriétés de couleur de la cible.

Soit $\mathbf{e}_t$ le vecteur formé par la concaténation au temps t de la position de l'objet $\bar{\mathbf{x}}_t = (x_t, y_t)^T$, de sa taille s_t et des composantes de vitesses u_t et v_t. La modélisation par filtrage particulaire [Pérez 02] utilise des particules pondérées afin d'échantillonner la distribution des états $p(\mathbf{e}_t|\mathbf{z}_{1:t})$, $\mathbf{z}_{1:t}$ étant l'ensemble des données observées dans les T images correspondant à l'intervalle de temps $[1:T]$. L'ensemble des M particules est noté $\{\mathbf{e}_t(i)\}$, $i = 1...M$, les poids correspondant étant $\omega_t(i)$. L'algorithme de filtrage peut alors être décrit par les étapes itératives suivantes :

◦ **étape de prédiction :** Propagation de l'ensemble des particules actuelles $\{\mathbf{e}_{t-1}(i)\}$ jusqu'à t à l'aide des dynamiques considérées.

◦ **étape de filtrage :** Comparaison des particules prédites aux mesures observées à l'aide de la fonction de vraisemblance $p(\mathbf{z}_t|\mathbf{e}_t^{(i)})$, et mise à jour des poids par $\omega_t^{(i)} \propto \omega_{t-1}^{(i)} p(\mathbf{z}_t|\mathbf{e}_t^{(i)})$ (où $\sum_i \omega_t^{(i)} = 1$).

◦ **étape de rééchantillonnage :** (optionnel) Tirage d'un nouvel ensemble de M particules, décrites par $\{\mathbf{e}_t(i), \omega_t^{(i)}\}$, avec des poids égaux.

4.1.2 Trajectoires vidéos

Des trajectoires fiables et précises ont, à l'aide des méthodes décrites dans la section précédente, été extraites de plans issus de vidéos de courses de Formule 1. La figure 4.1 présente des images issues de vidéos de courses de Formule1 d'où ont été extraites des trajectoires correspondant au suivi des véhicules. La figure 4.2 rassemble les classes de trajectoires ayant pu être extraites d'une vidéo de Formule1. Chacun des dix groupements de trajectoires correspond à une classe composée des trajectoires issues d'une même caméra placée sur le circuit automobile.

FIG. 4.1 – Trois images de deux plans vidéos (rangées du haut et du bas) filmés par deux caméras différentes dans un programme TV de Formule1 placées à deux endroits sur le circuit. Chaque ligne de la figure correspond à une séquence d'images associée à un plan donné. Les trajectoires recalées issues du suivi sont imprimées sur les images.

Comme le montre la figure 4.2, des différences, en termes de translation et d'échelle notamment, peuvent apparaître sur les trajectoires extraites. En effet, les vidéos traitées sont filmées par des caméras ayant des mouvement de rotation ainsi que des zooms. Alors, deux raisons peuvent expliquer les différences observées sur les trajectoires. La première est que les trajectoires sont recalées, dans la procédure de suivi, dans le plan image initial de la trajectoire (le plan image au temps t initial de la trajectoire), ce plan pouvant varier selon la position de la caméra au moment du début du suivi. La seconde cause de ces différences doit être associée à des erreurs de recalage lors de la procédure de compensation du mouvement de la caméra, erreurs caractérisées par des écarts et des biais en termes de translation ou d'échelle sur les trajectoires produites.

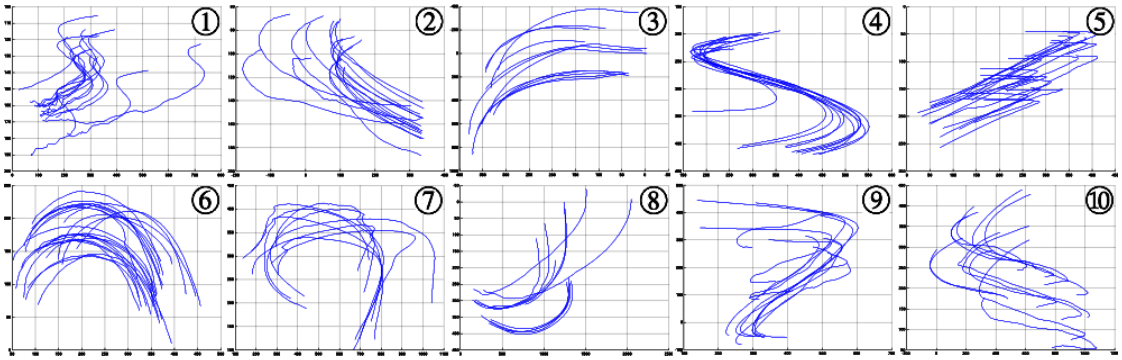

FIG. 4.2 – Tracé des 10 classes de trajectoires (149 trajectoires) extraites d'un programme TV de Formule1, classe par classe. Une classe est composée de trajectoires issues de plans vidéos filmés par une même caméra. Les différentes classes correspondent ainsi aux différentes caméras placées à des endroits stratégiques du circuit.

4.2 Représentation des trajectoires

Contrairement aux méthodes existantes pour l'analyse de trajectoires issues d'une même caméra (par exemple, pour la modélisation de parcours suivi par des piétons dans la surveillance d'un parking), notre but est de pouvoir traiter toutes les trajectoires, quelles que soient leurs longueurs et leurs provenances. Nous voulons extraire des classes correspondant à des mouvements similaires, en termes de forme des trajectoires et de vitesse de parcours, sans connaissance *a priori* sur la calibration des caméras et sur la structure de la scène ou sur le mouvement 3D de l'objet considéré. La méthode proposée doit également être capable de traiter des trajectoires issues d'une caméra unique comme de caméras différentes. Ainsi, afin de traiter des contenus issus de caméras différentes, nous proposons une représentation des trajectoires invariante à la translation 2D, à la rotation 2D et au facteur d'échelle. Ces invariances pourront également permettre de prendre en compte les différences observées, en termes de translation et d'échelle notamment, qui apparaissent après recalage lorsque l'on traite des vidéos filmées par des caméras mobiles (voir section 4.1).

4.2.1 Choix de la méthode d'approximation

Supposons qu'une trajectoire T_k soit définie par un ensemble de n_k points correspondant aux positions successives de l'objet suivi dans la séquence d'images, on note $T_k = \{(x_1, y_1), .., (x_{n_k}, y_{n_k})\}$. Afin de calculer les descripteurs $\dot{\gamma}(t)$ de cette trajectoire, qui sont des valeurs différentielles, il est souhaitable d'avoir une approximation continue (u_t, v_t) des courbes 2D formées par les trajectoires. Nous avons effectué une approximation par noyaux gaussiens des coordonnées en x et y de T_k (voir figure 4.3),

approximation nécessitant le choix de h qui est le paramètre de lissage de la courbe. Nous avons :

$$u_t = \frac{\sum_{j=1}^{n_k} e^{-(\frac{t-j}{h})^2} x_j}{\sum_{j=1}^{n_k} e^{-(\frac{t-j}{h})^2}},\ v_t = \frac{\sum_{j=1}^{n_k} e^{-(\frac{t-j}{h})^2} y_j}{\sum_{j=1}^{n_k} e^{-(\frac{t-j}{h})^2}}. \tag{4.1}$$

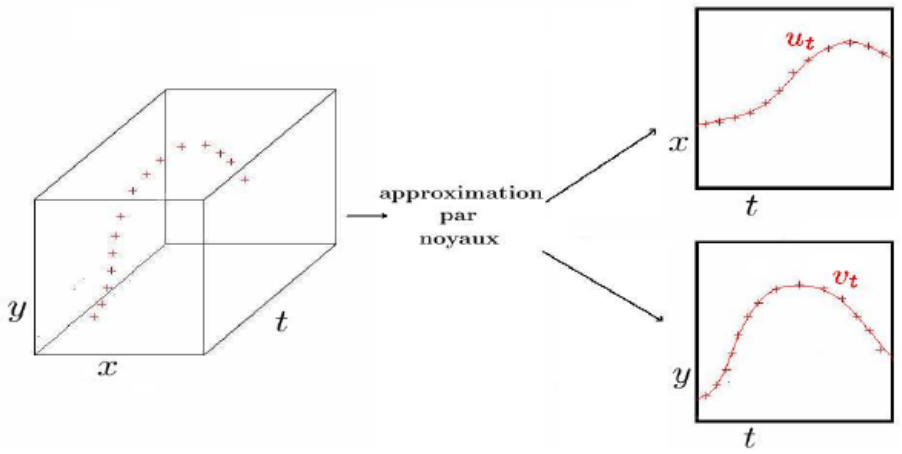

FIG. 4.3 – Schéma présentant une trajectoire ainsi que les approximations des coordonnées u_t et v_t obtenues à l'aide d'une approximation par noyaux.

Les expressions explicites $\dot{u}_t$, $\dot{v}_t$, $\ddot{u}_t$ et $\ddot{v}_t$ correspondant aux dérivées temporelles à l'ordre un et deux de u et v peuvent alors être obtenues par les formules classiques de dérivation.

4.2.2 Nature de la représentation

La plupart des méthodes pour la classification de trajectoires développées jusqu'à maintenant utilisent les coordonnées spatiales des points de la trajectoire dans l'image (voir section 3.2.1). Ces coordonnées sont en effet utiles pour étudier les ressemblances exactes entre trajectoires, mais notre approche est d'exploiter l'aspect général des trajectoires. Prendre en compte les orientations successives des trajectoires est plus intéressant, cela permettant d'avoir des invariances à la translation, à la rotation ainsi qu'au facteur d'échelle. De plus, il est alors possible de comparer les formes générales des trajectoires. Fashandi et al. [Fashandi 05] ont ainsi utilisé les différences relatives entre orientations locales.

Nous considérons tout d'abord les orientations locales des trajectoires données par $\gamma_t = \arctan(\dot{v}_t/\dot{u}_t)$, où u_t et v_t sont les coordonnées temporelles des trajectoires et $\dot{u}_t$ et $\dot{v}$ leurs dérivées temporelles. En effet, ces primitives fournissent une représentation invariante à la translation ainsi qu'au facteur d'échelle. Pour obtenir une représentation également invariante aux rotations, nous considérons en fait, plutôt que les différences

relatives entre orientations successives utilisées dans [Fashandi 05], sa dérivée temporelle $\dot{\gamma}_t$. Les propriétés d'invariance de la dérivée temporelle $\dot{\gamma}_t$ seront développées ci-dessous. Pour ce faire, il est utile d'exprimer $\dot{\gamma}_t$ en fonction des dérivées temporelles de u_t et v_t. Ainsi, on peut montrer que [Hervieu07a] :

$$\tan\gamma_t = \frac{\dot{v}_t}{\dot{u}_t}, \text{ alors } \frac{d(\tan\gamma_t)}{dt} = \frac{\ddot{v}_t\dot{u}_t - \ddot{u}_t\dot{v}_t}{\dot{u}_t^2}.$$

Par ailleurs,

$$\frac{d(\tan\gamma_t)}{dt} = \frac{1}{\cos^2\gamma_t}\dot{\gamma}_t,$$

ainsi, on a :

$$\dot{\gamma}_t = \cos^2\gamma_t \left(\frac{\ddot{v}_t\dot{u}_t - \ddot{u}_t\dot{v}_t}{\dot{u}_t^2}\right).$$

Or,

$$1 + \tan^2\gamma_t = \frac{1}{\cos^2\gamma_t} \text{ et ainsi } \frac{1}{\cos^2\gamma_t} = 1 + \frac{\dot{v}_t^2}{\dot{u}_t^2} \text{ et } \cos^2\gamma_t = \frac{\dot{u}_t^2}{\dot{u}_t^2 + \dot{v}_t^2}$$

et donc finalement,

$$\dot{\gamma}_t = \frac{\ddot{v}_t\dot{u}_t - \ddot{u}_t\dot{v}_t}{\dot{u}_t^2 + \dot{v}_t^2} = \kappa_t.\|V_t\|, \tag{4.2}$$

avec $\kappa_t = \frac{\ddot{v}_t\dot{u}_t - \ddot{u}_t\dot{v}_t}{(\dot{u}_t^2+\dot{v}_t^2)^{\frac{3}{2}}}$ la courbure algébrique locale de la trajectoire et $\|V_t\| = (\dot{u}_t^2 + \dot{v}_t^2)^{\frac{1}{2}}$ l'amplitude de la vitesse locale au point (u_t, v_t) dans le plan image. Cette représentation permet donc, en plus de ses propriétés intrinsèques d'invariance, d'avoir une prise en compte combinée d'informations de forme et de dynamique (en terme d'évolution de la vitesse), ce qui pourra s'avérer déterminant pour l'interprétation des trajectoires vidéos.

- ***Propriété d'invariance de la représentation***

Les orientations locales γ_t sont évidemment invariantes à la translation ainsi qu'au facteur d'échelle, dans le plan image. La dérivée temporelle de γ_t, $\dot{\gamma}_t$, est donc également invariante à la translation et au facteur d'échelle. De plus, le numérateur de $\dot{\gamma}_t$, $\ddot{v}_t\dot{u}_t - \ddot{u}_t\dot{v}_t = \det\begin{pmatrix} \ddot{v}_t & \dot{v}_t \\ \ddot{u}_t & \dot{u}_t \end{pmatrix}$ est un déterminant, donc invariant à la rotation 2D. Il en est de même du dénominateur de $\dot{\gamma}_t$, $\dot{u}_t^2 + \dot{v}_t^2 = \|V_t\|^2$, invariant aux rotations 2D puisqu'il correspond à la norme de la vitesse. Par conséquent $\dot{\gamma}_t$ est invariant aux rotations 2D, et donc $\dot{\gamma}(t)$ est bien invariant aux transformations de type translation, rotation, échelle ainsi que leurs compositions, dans le plan image.

Le vecteur de descripteurs utilisé pour représenter une trajectoire T_k de taille n_k, sera alors le vecteur ϕ contenant les valeurs successives de $\dot{\gamma}(t)$:

$$\phi = [\dot{\gamma}_1, \dot{\gamma}_2, ..., \dot{\gamma}_{n_k-1}, \dot{\gamma}_{n_k}]. \tag{4.3}$$

4.2.3 Choix du paramètre de lissage

Afin d'avoir une méthode automatique de calcul des primitives représentant les trajectoires, le choix de la valeur du paramètre h, dans l'équation 4.1, doit être résolu. Ainsi, pour une trajectoire T_k de taille n_k, un critère d'erreur quadratique moyenne (EQM) est considéré [Härdle 04], défini par

$$EQM(h) = EQM(\hat{u}_{t,h}) = \frac{1}{n_k}\sum_{i=1}^{n_k}(\hat{u}_{i,h} - u_i)^2,$$

où les u_i correspondent aux valeurs vraies (mais inconnues) des coordonnées à estimer, et $\hat{u}_{i,h}$ sont leurs estimées. Une approximation naïve $p(h)$ de $EQM(h)$, ou "estimation par substitution", est faite en remplaçant u_i par les coordonnées observées x_i :

$$p(h) = \frac{1}{n_k}\sum_{i=1}^{n_k}(x_i - \hat{u}_{i,h})^2. \tag{4.4}$$

En additionnant et soustrayant u_i à (4.4), on obtient

$$\begin{aligned} p(h) &= \frac{1}{n_k}\sum_{i=1}^{n_k}((x_i - u_i) + (u_i - \hat{u}_{i,h}))^2 \\ &= \frac{1}{n_k}\sum_{i=1}^{n_k}\varepsilon_i^2 + ASE(h) - \frac{2}{n_k}\sum_{i=1}^{n_k}\varepsilon_i(\hat{u}_{i,h} - u_i) \end{aligned} \tag{4.5}$$

où $\varepsilon_i = x_i - u_i$, et $EQM(h) = \frac{1}{n_k}\sum_{i=1}^{n_k}(u_i - \hat{u}_{i,h})^2$ est l'erreur quadratique moyenne sur les coordonnées. En considérant maintenant une méthode de validation croisée,

$$VC(h) = \frac{1}{n_k}\sum_{i=1}^{n_k}(x_i - \hat{u}_{-i,h})^2 \tag{4.6}$$

où $\hat{u}_{-i,h}$ est un estimateur "leave-one-out" donné par

$$\hat{u}_{-i,h} = \frac{\sum_{j\neq i} e^{-(\frac{i-j}{h})^2}x_j}{\sum_{j\neq i} e^{-(\frac{i-j}{h})^2}},$$

on montre que l'espérance du dernier terme de (4.5) est zéro si, à la place de $\hat{u}_{i,h}$, on considère $\hat{u}_{-i,h}$, *i.e.* :

$$\mathbb{E}[-\frac{2}{n_k}\sum_{i=1}^{n_k}\varepsilon_i(\hat{u}_{-i,h} - u_i)] = 0.$$

De plus, le premier terme de (4.5), $\frac{1}{n_k}\sum_{i=1}^{n_k}\varepsilon_i^2$, est indépendant de h. Ainsi, choisir h_{opt}tel que $VC(h)$ est minimisé revient donc à minimiser, en moyenne, l'erreur quadratique moyenne $EQM(h)$, permettant ainsi au mieux de prendre en compte le bruit

inhérent aux coordonnées des trajectoires (ici, les calculs ont été développés pour u, et sont appliqués de la même manière pour v). En pratique, h définissant une fenêtre de lissage sur les données, les valeurs de h testées pour $VC(h)$ sont supérieures à l'écart entre deux points successifs (on prendra $h \geq 1$).

4.2.4 Illustrations

Nous décrivons, dans cette section, quelques expérimentations liées à la représentation ainsi qu'à son extraction. Ces illustrations permettent de mettre en valeur les propriétés de la représentations choisie ainsi que des procédures d'extraction décrites ci-dessus.

4.2.4.1 Propriétés de la représentation choisie

Dans cette section, nous présentons les distributions de la primitive choisie (*i.e.*, les valeurs $\dot{\gamma}$) obtenues, pour des classes de trajectoires synthétiques (*i.e.*, de trajectoires générées informatiquement à l'aide des formulations paramétriques connues) de formes variées comprenant (voir figure 4.4) :

- des sinusoides,
- des paraboles,
- des hyperboles,
- des ellipses,
- des cycloides,
- des spirales,
- des droites,
- des clothoïdes.

La figure 4.4 permet de mettre en valeur, expérimentalement, les propriétés de la représentation choisie. Tout d'abord, en termes d'invariances, nous avons appliqué à des trajectoires diverses des transformations de translations, de rotations et d'échelle dans le plan image sans avoir aucun effet sur les valeurs calculées de $\dot{\gamma}$, permettant de confirmer les calculs et les observations effectuées en section 4.2.2.

De plus, pour produire les classes de trajectoires, nous avons utilisé les formes paramétriques qui leur sont associées, de la forme :

$$\begin{cases} x(t) = f(\theta_x, t) \\ y(t) = g(\theta_y, t) \end{cases}$$

Ainsi, nous avons créé, pour chaque classe synthétique, un corpus de trajectoires avec différents paramètres θ_x et θ_y, et ce afin de pouvoir comparer les distributions de la variable aléatoire $\dot{\gamma}$ pour des trajectoires similaires (en termes de formes) mais ayant des paramétrisations différentes. Ces différents choix de paramétrisations permettent tout d'abord d'avoir des trajectoires d'une même classe (*i.e.*, d'une même forme) avec des

caractéristiques différentes (par exemple, différentes fréquences pour les sinusoides). Ils permettent également de faire varier les vitesses de parcours le long des trajectoires. La figure 4.4, qui présente deux trajectoires par classes choisies aléatoirement ainsi que les histogrammes des $\dot{\gamma}$ correspondants, met en valeur ce que nous avons observé expérimentalement. En effet, les distributions des $\dot{\gamma}$ associées aux trajectoires d'une même classe s'avèrent assez similaires (en général, très similaires, bien que les deux spirales aient une distribution de $\dot{\gamma}$ sensiblement différentes dans la figure 4.4). De plus, on peut observer que des courbes à l'allure proche (les hyperboles et les paraboles par exemple) ont des distributions de $\dot{\gamma}$ également proches (voir les troisième et quatrième lignes de la figure 4.4). Ces premières observations montrent donc que la représentation choisie doit pouvoir permettre de distinguer les différentes trajectoires rencontrées (en terme de dynamique et de forme) et donc servir pour des tâches de reconnaissance.

Il est à noter dans la figure 4.4, que toutes les distributions sont présentées volontairement dans l'intervalle $[-2,\ 2]$, mais que cette "étendue" dans les distributions des primitives associées aux courbes dépend de la vitesse de parcours de la trajectoire. En effet, le paramètre t des formules de paramétrisation correspond au temps, et, donc, aux images des vidéos. Ainsi, ce paramètre agit sur la vitesse apparente de parcours des différentes trajectoires. En changeant l'échantillonnage temporel associé d'une trajectoire donnée, l'ordre de grandeur des valeurs de $\dot{\gamma}$ observées est modifié (par exemple, la distribution de $\dot{\gamma}$ évoluera non plus dans $[-2,\ 2]$, mais dans un autre intervalle), néanmoins, la forme générale des distribution de $\dot{\gamma}$ n'est pas altérée. Cela était prévisible puisque

$$\dot{\gamma}_t = \kappa_t.\|V_t\|,$$

où $\|V_t\|$ correspond à l'amplitude de la vitesse locale. La forme, traduite par le terme de courbure κ_t, ne change pas quelle que soit la vitesse de parcours sur la trajectoire. L'amplitude des valeurs de $\dot{\gamma}$, pour une même trajectoire, est donc une fonction linéaire de la vitesse de parcours des objets.

Un dernier point à souligner est le signe des valeurs de $\dot{\gamma}$. Comme le montre la figure 4.4, les valeurs de $\dot{\gamma}$ peuvent être négatives ou positives selon le sens de parcours de la trajectoire. $\dot{\gamma}_t = \kappa_t.\|V_t\|$ et, donc, le signe de $\dot{\gamma}$ est fonction du signe κ_t, la courbure à l'instant t. Le signe de la courbure s'interprète en fait comme l'indication du sens dans lequel est tournée la concavité de la courbe. Une interprétation de cette dernière observation est la suivante, si la tangente à la courbe "coupe" la trajectoire et passe de l'autre coté de la courbe, alors la courbure change de signe. Si la trajectoire se trouve dans la portion concave définie par la courbe, alors la courbure est positive.

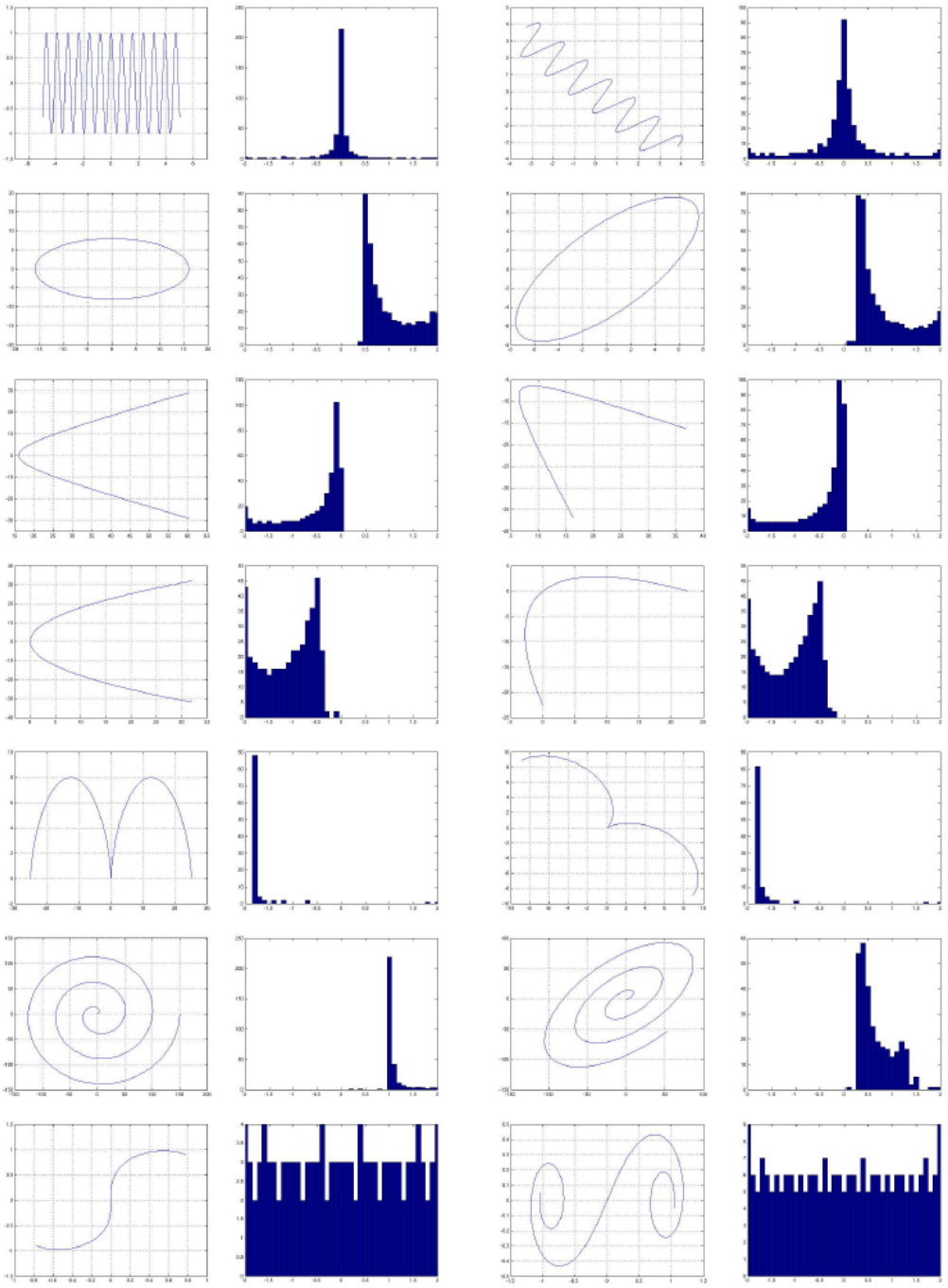

FIG. 4.4 – Pour chaque classe de formes (sinusoïdes, paraboles, hyperboles, ellipses, cycloïdes, spirales et clothoïdes) sont tracées deux courbes ayant des paramétrisations différentes et présentant des différences en translation, orientation et facteur d'échelle, et les histogrammes de $\dot{\gamma}$ correspondants.

4.2.4.2 Sélection du paramètre de lissage

Nous présentons dans ce paragraphe des résultats relatifs à la méthode de sélection du paramètre h. De nombreuses méthodes ont été proposées pour le choix des paramètres de lissage dans les méthodes de régressions à noyaux. La procédure que nous avons choisi, s'avère en fait la plus simple à mettre en œuvre, tout en assurant un cadre théorique solide (voir section 4.2.1 et [Härdle 04]). Elle repose sur une technique de validation croisée "leave-one-out", caractérisée par la valeur $VC(h)$ (définis dans la relation 4.6). $VC(h)$ est calculé pour les valeurs de h ($h > 1$, voir section 4.2.1), la valeur $\tilde{h}$ retenue étant celle qui minimise $VC(h)$.

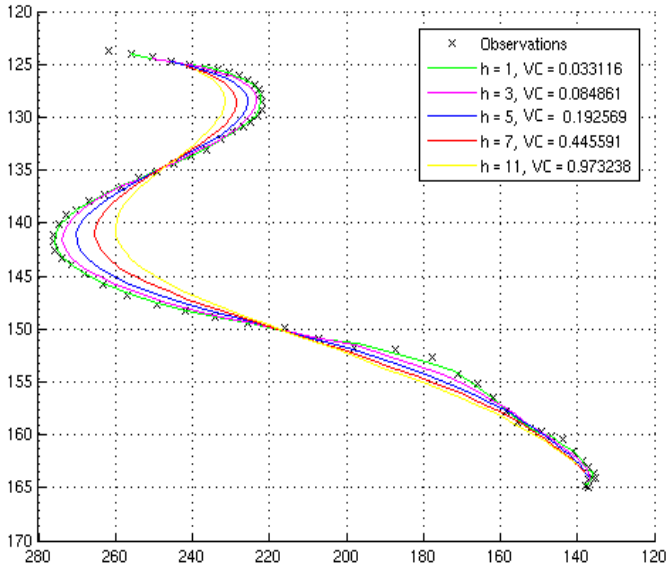

FIG. 4.5 – Une séquence de coordonnées issue de suivi dans des vidéos de Formule1 et les courbes lissées correspondantes, pour certaines valeurs du paramètre h.

Les figures 4.5 et 4.6 contiennent, pour deux séquences d'observations correspondant à des trajectoires extraites de vidéo de Formule1, les valeurs de $VC(h)$ pour plusieurs valeurs de h ainsi que les courbes "lissées" correspondantes. La figure 4.5 présente une trajectoire assez lisse, et la valeur sélectionnée est $\tilde{h} = 1$ (quand les observations sont peu bruitées, un h de faible valeur est choisi). La figure 4.6 correspond, elle, à une trajectoire artificiellement bruitée, à l'aide d'un bruit blanc appliqué sur les valeurs en abscisses et en ordonnées des points de la trajectoire originelle. Dans ce cas, la méthode statistique de sélection du paramètre de lissage h décrite en section 4.2.1 retient une valeur de h sensiblement plus grande (ici, $\tilde{h} = 6$). Les valeurs de h sélectionnées sont fonction du bruit inhérent aux observations.

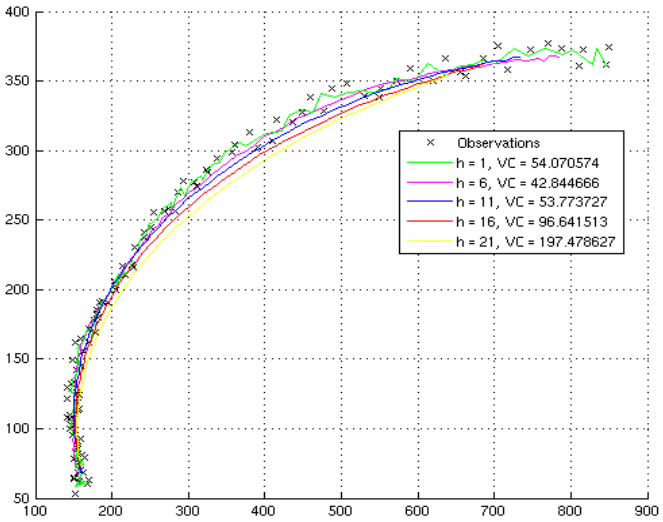

FIG. 4.6 – Une séquence de coordonnées issue de suivi dans des vidéos de Formule1 bruitées artificiellement, et les courbes lissées correspondantes, pour certaines valeurs du paramètre h.

4.3 Conclusion

Nous avons introduit une représentation des trajectoires vidéos à l'aide de primitives, notées $\dot{\gamma}$, correspondant à la dérivée temporelle de l'orientation locale des trajectoires. Cette représentation a deux propriétés importantes dans le cadre du traitement de trajectoires vidéos. La première est son invariance à des transformations de translation, de rotation et d'échelle dans le plan image. La seconde propriété des primitives considérées, désirée et obtenue, est la prise en compte combinée d'informations de dynamique (vitesse de parcours de la trajectoire) et de forme (courbure) associées aux trajectoires traitées. Une première comparaison des distributions de primitives pour différentes classes "typiques" de courbes a permis de mettre en valeur les bonnes capacités de caractérisation de la représentation choisie. Elle pourra, comme on le montrera dans la suite de ce manuscrit, être utilisée dans des tâches de reconnaissance de trajectoires dans des vidéos. De plus, la technique de sélection du paramètre de la méthode de calcul des primitives, pour la prise en compte du bruit contenu dans les trajectoires vidéos extraites, s'est montrée opérante.

Chapitre 5

Reconnaissance d'évènements à l'aide de trajectoires

"L'imagination, l'illumination, la création, sans lesquelles le progrès des sciences n'aurait pas été possible, n'entraient dans la science qu'en catimini : elles n'étaient pas logiquement repérables, et toujours épistémologiquement condamnables. On en parlait dans les biographies des grands savants, jamais dans les manuels et les traités, dont pourtant la sombre compilation, comme les couches souterraines de charbon, était constituée par la fossilisation et la compression de ce qui, au premier chef, avait été fantaisies, hypothèses, prolifération d'idées, inventions, découvertes.
Effectivement, la part à la fois gravide et lourde, éthérée et onirique de la réalité humaine (et peut-être de la réalité du monde) a été prise en charge par l'irrationnel, part maudite, part bénie où la poésie gorgeait et dégorgeait ses essences ; qui, filtrées et distillées un jour, pourraient et devraient s'appeler science."

Edgar Morin - ***Introduction à la pensée complexe***

Comme nous l'avons présenté dans le chapitre précédent, une représentation adaptée, à l'aide des primitives $\dot{\gamma}$, a été proposée pour le traitement de trajectoires extraites de séquences vidéos pouvant être acquises par des caméras mobiles. L'objectif est maintenant d'exploiter les propriétés de cette caractérisation de trajectoires pour différents objectifs de vision par ordinateur. À cette fin, nous avons développé une modélisation originale, à l'aide de MMC permettant la prise en compte des causalités temporelles.

Un de ses avantages est de pouvoir modéliser efficacement le contenu causal de "petits" ensembles de données, et, donc, des trajectoires de petites tailles qui sont souvent rencontrées dans des vidéos. Les plans vidéos dont sont extraites les trajectoires peuvent être, comme on le verra dans la suite, des plans de courtes durées. Ainsi, les trajectoires qui en sont extraites sont des séquences de coordonnées réduites à quelques dizaines d'observations. Afin de pouvoir comparer ces trajectoires, il est alors important d'avoir une modélisation pouvant efficacement exploiter ces ensembles réduits de données. Les MMC proposés, au contraire de ceux reposant sur des algorithmes d'apprentissage complexes, permet de modéliser les ensembles de données de faibles tailles.

La sélection des paramètres de cette modélisation, et notamment le nombre d'états des MMC considérés, a fait l'objet d'un soin tout particulier. Différentes tâches d'analyse de vidéos ont été développées, du clustering de contenus à la détection d'évènements inattendus, incluant la reconnaissance supervisée de plans vidéos à l'aide de leurs contenus dynamiques décrits par les trajectoires d'objets mobiles.

5.1 Modélisation de trajectoires par modèles de Markov cachés

Cette section décrit la modélisation par MMC développée pour le traitement de trajectoires vidéos. Le but de la modélisation par MMC proposée tient en deux points : le premier est d'appréhender les causalités temporelles inhérentes aux trajectoires observées, le second est de comparer des trajectoires potentiellement de petites tailles (pour des tâches de clustering par exemple).

Les états des MMC proposés correspondent aux bins d'une quantification sur les observations. Le nombre de ces états est automatiquement choisi et justifié à l'aide d'une méthode statistique originale. Elle peut être plus généralement utilisée pour des choix de nombre de bins pour des tâches de classification à l'aide d'histogrammes. Dans la suite de ce document, les MMC développés seront dénotés par MMCQ (pour MMC par Quantification) [Hervieu 08c].

5.1.1 Modèles de Markov cachés pour la modélisation d'ensembles de données restreints

La prise en compte de la causalité temporelle inhérente aux trajectoires issues de vidéos est réalisée à l'aide de MMCQ. Les états associés à la modélisation par MMCQ sont définis, de façon originale, par les bins d'une quantification effectuée sur les descripteurs $\dot{\gamma}$. Cette hypothèse doit permettre d'éviter le recours à des algorithmes de partitionnement "lourds" à entraîner (tel que l'algorithme *EM*, voir section 2.3), nécessitant un nombre important d'observations. En effet, les trajectoires rencontrées sont souvent de faibles tailles et ne permettent pas un entraînement efficace de tels

algorithmes. Avec la modélisation par MMCQ, les états des modèles sont directement définis grâce à une quantification de la distribution des observations. Le MMCQ permet alors de modéliser les causalités temporelles entre les observations successives selon les bins auxquelles elles appartiennent.

Afin de déterminer les états des MMCQ, la distribution des $\dot{\gamma}$ de chaque trajectoire est considérée. Pour une trajectoire T_k, nous avons choisi de fixer un intervalle $[B_{1,k}, B_{2,k}]$ autour de la valeur moyenne m_k des $\dot{\gamma}$ observés et contenant un certain pourcentage P_v des valeurs de $\dot{\gamma}$ mesurées (pour la trajectoire considérée). De façon empirique, nous avons constaté qu'une valeur de P_v autour de 95% donnait les résultats les plus concluants, ainsi dans la suite, P_v sera pris égal à 95%.

Ensuite, l'espace des observations est divisé en un nombre N_k de bins. Ces bins correspondent aux états du MMCQ, le bin i (ou état i) sera noté S_i dans la suite. L'intervalle $[B_{1,k}, B_{2,k}]$ est tout d'abord divisé en N'_k bins appelés les états intérieurs. Deux états extrêmaux (non bornés) S_1 et S_{N_k} sont également mis en place, définis par $]-\infty, B_{1,k}]$ et $[B_{2,k}, +\infty[$. Les états des MMCQ associés aux trajectoires seront donc les $N_k = N'_k + 2$ bins de la quantification. Ceci est illustré par la figure 5.1, qui présente une quantification des P_v pourcent (*i.e.*, dans $[B_{1,k}, B_{2,k}]$) des $\dot{\gamma}$ observés.

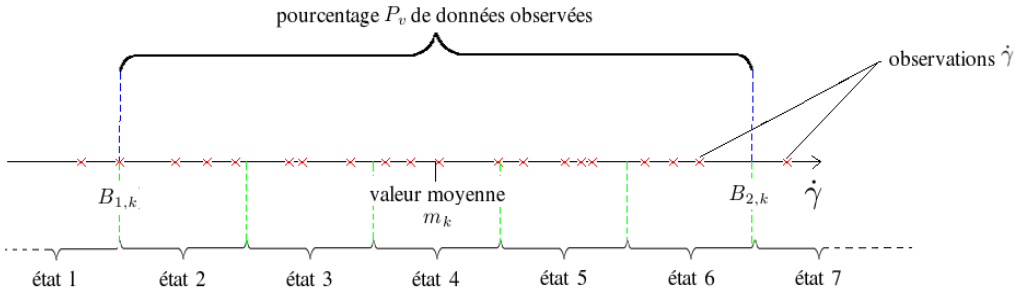

FIG. 5.1 – Quantification effectuée sur les descripteurs calculés $\dot{\gamma}$ associés à une trajectoire T_k, avec cinq bins correspondant aux états intérieurs ($N_k = 7$).

Le MMCQ modélisant une trajectoire T_k est caractérisé par :
- la matrice de transition entre états $A = \{a_{ij}\}$ avec

$$a_{ij} = P[\, q_{t+1} = S_j \mid q_t = S_i \,], \;\; 1 \leq i, j \leq N_k,$$

où q_t est la variable d'état à l'instant t et S_i sa valeur (*i.e.*, le i^{me} bin de l'histogramme) ;
- la distribution initiale des états $\pi = \{\pi_i\}$, où $\pi_i = P[\, q_1 = S_i \,], \;\; 1 \leq i \leq N_k$;
- les probabilités d'observations conditionnelles $B = \{b_i(\dot{\gamma}_t)\}, \;\; 1 \leq i \leq N_k$, où $b_i(\dot{\gamma}_t) = P[\dot{\gamma}_t \mid q_t = S_i]$.

Dans la modélisation choisie, les probabilités d'observation conditionnelles $\tilde{P}[\dot{\gamma}_t \mid q_t = S_i]$ sont tout d'abord définies, dans $[B_{1,k}, B_{2,k}]$, par des distributions gaussiennes

de moyennes μ_i (*i.e.*, la valeur médiane du bin S_i). Leurs variances σ ne dépendent pas de l'état considéré mais sont spécifiées de telle façon que les intervalles $[\mu_i - \sigma, \mu_i + \sigma]$ correspondent aux tailles des bins. La figure 5.2 illustre ce choix de probabilité d'observation conditionnelle. L' intervalle $[\mu_i - \sigma, \mu_i + \sigma]$ a été choisi car il contient 95% des observations d'une gaussienne, pourcentage régulièrement utilisé pour les tests statistiques. De plus, nous avons comparé ce choix en considérant, pour correspondre aux tailles des bins, les intervalles $[\mu_i - 2\sigma, \mu_i + 2\sigma]$, $[\mu_i - 3\sigma, \mu_i + 3\sigma]$ et $[\mu_i - \frac{\sigma}{2}, \mu_i + \frac{\sigma}{2}]$. Les meilleurs résultats pour les tâches développées ultérieurement ont bien été obtenus avec le choix de l'intervalle $[\mu_i - \sigma, \mu_i + \sigma]$ égal à la taille des bins.

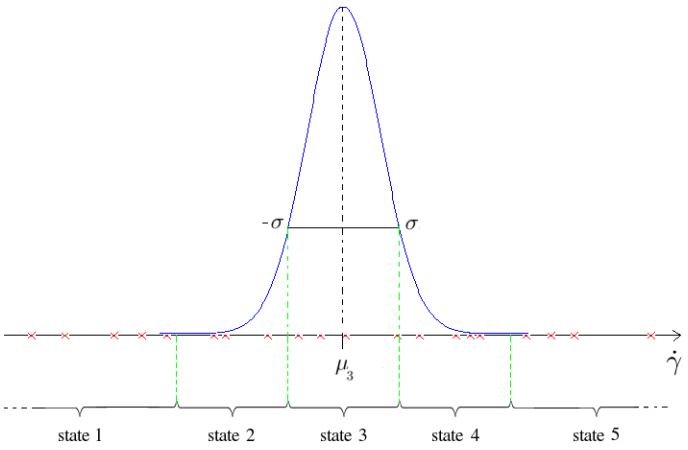

FIG. 5.2 – Illustration du choix de probabilités d'observations conditionnelles, l'intervalle $[\mu_3 - \sigma, \mu_3 + \sigma]$ correspondent à la taille du bin S_3.

Ces probabilités d'observations conditionnelles sont ensuite normalisées de sorte que, pour toute observation $\dot{\gamma}$, $\sum_{i=1...N_k} P[\dot{\gamma}_t \mid q_t = S_i] = 1$. Ainsi, les probabilités d'observations conditionnelles sont finalement données par :

$$P[\dot{\gamma}_t \mid q_t = S_i] = \frac{\tilde{P}[\dot{\gamma}_t \mid q_t = S_i]}{\sum_{i=1...N_k} P[\dot{\gamma}_t \mid q_t = S_i]}.$$

En dehors de $[B_{1,k}, B_{2,k}]$, les observations appartiennent à l'état extrême correspondant. Une illustration des probabilités d'observations conditionnelles obtenues de cette manière est donnée dans la figure 5.3.

Estimation des paramètres du modèle

Afin d'estimer les paramètres du modèle A et π, nous avons adopté la méthode par moindres carrés définie dans [Ford 98] en assimilant les MMCQ à un processus de comptage. Alors, les valeurs $b_i(\dot{\gamma}_t) = P(\dot{\gamma}_t | q_t = S_i)$ (correspondant à un poids pour

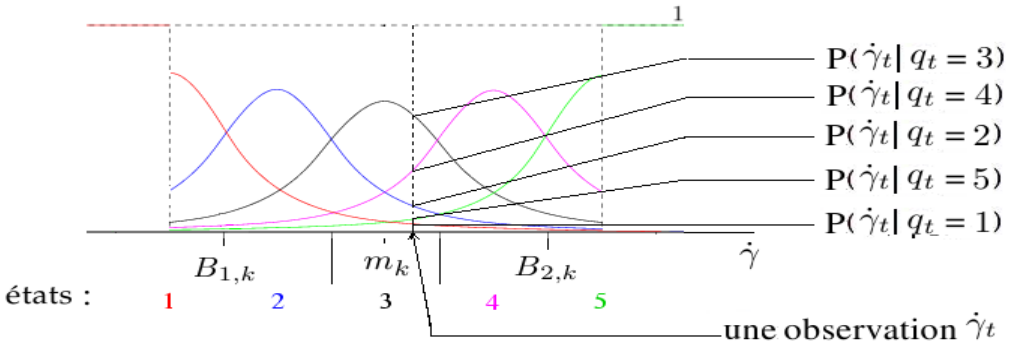

FIG. 5.3 – Illustration des probabilités d'observations conditionnelles pour une trajectoire T_k, avec un nombre d'états $N_k = 5$.

le processus de comptage effectué), sont utilisées afin d'estimer les valeurs de A et π, pour $1 \leq i \leq N_k$ et $1 \leq j \leq N_k$, par :

$$a_{ij} = \frac{\sum_{t=1}^{n_k-1} b_i(\dot{\gamma}_t) b_j(\dot{\gamma}_{t+1})}{\sum_{t=1}^{n_k-1} b_i(\dot{\gamma}_t)} \text{ et } \pi_i = \frac{\sum_{t=1}^{n_k} b_i(\dot{\gamma}_t)}{n_k}, \tag{5.1}$$

où n_k correspond aux nombre d'observations associé à la trajectoire T_k (*i.e.*, la taille de la trajectoire T_k).

Ainsi, une fois la quantification des valeurs $\dot{\gamma}$ effectuée, l'ensemble des paramètres du MMCQ, pour une trajectoire T_k, est directement obtenu. Au contraire, les algorithmes de type *EM*, qui estiment les paramètres des modèles de façon itérative (en général, de mélanges de distributions), requièrent un nombre d'observations important [Porikli 04a]. Ce modèle de probabilités d'observations conditionnelles, en dépit de sa simplicité, a un avantage notable. En effet, ce choix exprimé sur l'ensemble des observations associé à une trajectoire permet d'avoir une modélisation ne nécessitant pas de contraintes sur le nombre d'observations nécessaire (et, donc, pouvant traiter des ensembles d'observations de petites tailles), et ce tout en tenant compte du caractère causal des trajectoires.

La figure 5.4 présente trois exemples de trajectoires vidéos, leurs représentations lissées, les histogrammes correspondants ainsi que les matrices de transition A et les distributions initiales des états π estimé.

F. Porikli a proposé une méthode de classification basée sur les MMC permettant de modéliser la causalité temporelle des trajectoires étudiées [Porikli 04a]. Néanmoins, il s'est appuyé sur une modélisation par MMC continus classiques avec une topologie de type "left-to-right" (voir section 1.3.1) et des probabilités d'observation conditionnelle formalisées par des mélanges de gaussiennes (section 1.3.1). Cette modélisation

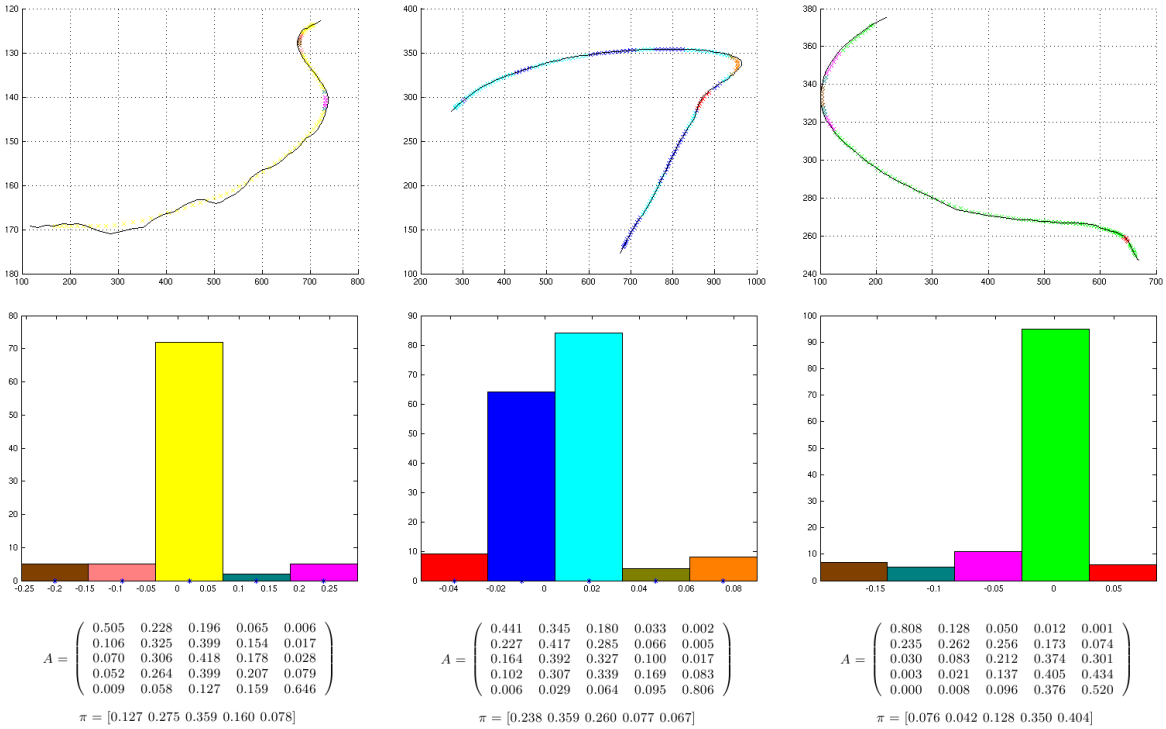

FIG. 5.4 – Partie supérieure : tracé des trajectoires réelles extraites de vidéos de Formule1 et leurs représentations lissées. Les couleurs des points des trajectoires sont associées aux différents états de la quantification et correspondent aux couleurs des bins des histogrammes associés. Partie médiane : histogrammes correspondant aux trois trajectoires. Les couleurs associées expriment les différentes valeurs d'état du MMCQ (les couleurs ont été choisies de façon aléatoire, les états considérés étant différents pour chaque trajectoire). Partie inférieure : matrice de transition A et distribution initiale des états π estimés associés aux trois trajectoires présentées.

nécessite, pour chaque trajectoire, le choix du nombre d'états et l'apprentissage du modèle. Ces tâches qui s'avèrent complexes et peu fiables, comme a pu le souligner F. Porikli, lorsque l'on traite des trajectoires de petites tailles. Dans [Porikli 04a], il précise qu'afin d'avoir un apprentissage des modes du mélange (*i.e.*, dans la méthode mise en place, des gaussiennes) suffisamment précis, la longueur des trajectoires traitées, notées N doit vérifier

$$N \gg M \times K$$

où K est le nombre d'état du modèle et M le nombre de modes associé à chaque état (supposé constant pour chaque état). Ainsi, pour un MMC "left-to-right" comprenant, par exemple, quatre états représentés par un mélange de trois gaussiennes, la taille minimum des trajectoires prise en compte est de 120 points. Or, le modèle a une architecture "left-to-right" simplifiant l'apprentissage du MMC. Ainsi, cette méthode est sensible aux tailles des trajectoires, ce qui signifie que des trajectoires de tailles très différentes seront considérées comme différentes. La taille minimum requise pour un MMC ergodique, qui permettrait de comparer de façon pertinente des trajectoires de tailles différentes, serait donc plus importante. La méthode par MMCQ proposée qui ne prend pas en compte les tailles des trajectoires comme élément distinctif et ne nécessite pas (comme on pourra le souligner dans les expérimentations du chapitre suivant) de telles contrainte sur la taille des trajectoires observées.

En effet, afin de pouvoir effectuer la reconnaissance de contenu dans des plans vidéos issues de caméras différentes (et donc, filmant des scènes différentes) à l'aide de trajectoires vidéos, la modélisation proposée est insensible aux tailles des trajectoires. Certains contenus dynamiques équivalents peuvent avoir une durée plus ou moins variable, et donc des trajectoires plus ou moins longues. Par exemple, dans le chapitre suivant, nous tenterons d'effectuer des reconnaissances de contenu dans des plans issues de vidéos de ski. Deux classes à considérer peuvent alors être la classe "slalom" et la classe "descente", chacune filmée par différentes caméras pour des courses différentes. Une méthode désirant réaliser un clustering de telles classes de plans vidéos ne doit pas prendre en compte les tailles des trajectoires extraites comme élément distinctif. Différentes caméras filment les skieurs pendant des périodes différentes, selon leurs positions et selon la course. L'invariance à la longueur des trajectoires est alors un élément nécessaire.

La section suivante proposant une méthode statistique pour la sélection du nombre d'états devant être utilisés avec les MMCQ.

5.1.2 Choix du nombre d'états des MMCQ

Dans la modélisation décrite dans la sous-section précédente, le nombre d'états intérieurs N'_k introduits dans la quantification (dans $[B_{1,k}, B_{2,k}]$) pour une trajectoire T_k reste à déterminer. Afin de fixer le choix sur ce nombre de bins (et, donc, d'état pour les MMCQ), un critère de décision statistique définissant un équilibre entre le

nombre d'états et la confiance dans les valeurs des histogrammes correspondants a été développée. Le nombre d'états doit être réduit (pour éviter une sur-représentation du modèle) tout en maximisant la confiance dans les valeurs associées à l'état correspondant (afin d'avoir une représentation fiable).

Il est à noter que tous les développements effectués dans le reste de cette section pour le choix du nombre d'états d'un MMCQ sont directement transposables au choix du nombre de bins dans tout histogramme.

Soit Θ_i la valeur vraie (inconnue) correspondant à l'état (ou bin) i de l'histogramme normalisé représentant la distribution des $\dot{\gamma}$ associés à une trajectoire T_k (de taille n_k), et soit $\hat{\Theta}_i$ son estimateur défini comme la proportion de $\dot{\gamma}$ observés dans l'état i :

$$\hat{\Theta}_i = \frac{K_i}{n'_k},$$

où :

$$K_i = \sum_{l=1}^{n'_k} X_{i,l} = \sum_{l=1}^{n'_k} \mathbb{1}_{\{\dot{\gamma}_l \in S_i\}}, i = 2 \dots N'_k + 1$$

est le nombre d'observation dans l'état S_i, et

$$n'_k = \sum_{i=2}^{N'_k+1} K_i$$

correspond au nombre total d'observations dans $[B_{1,k}, B_{2,k}]$.
$X_{i,l} = \mathbb{1}_{\{\dot{\gamma}_l \in S_i\}}$ est la fonction d'appartenance de $\dot{\gamma}_l$ dans les états intérieurs. L'hypothèse est faite que $X_{i,l}$ suit une loi de Bernoulli sur $[B_{1,k}, B_{2,k}]$.
Alors, en utilisant le théorème central limite [Feller 71],

$$\frac{\hat{\Theta}_i - \mathbb{E}[\hat{\Theta}_i]}{\sqrt{\mathbb{V}[\hat{\Theta}_i]}} \underset{n_k}{\overset{\mathcal{L}}{\rightarrow}} \mathcal{N}(0,1),\ \forall i = 2 \dots N'_k + 1.$$

$\hat{\Theta}_i$ est trivialement un estimateur non biaisé de Θ_i, de sorte que l'intervalle de confiance IC_{95} (avec un pourcentage de confiance de 95%) de Θ_i est défini par :

$$IC_{95}(\Theta_i) = [\hat{\Theta}_i - \alpha_{95}\mathbb{V}[\hat{\Theta}_i],\ \hat{\Theta}_i + \alpha_{95}\mathbb{V}[\hat{\Theta}_i]],$$

où le quantile α_{95} est la valeur assurant que :

$$P(\Theta_i \in \ \Big]\hat{\Theta}_i - \alpha_{95}\mathbb{V}[\hat{\Theta}_i],\ \hat{\Theta}_i + \alpha_{95}\mathbb{V}[\hat{\Theta}_i]\Big[) \geq 0.95.$$

La variable aléatoire $X_{i,l}$ suit une loi de Bernoulli, ainsi :

$$\mathbb{V}[X_{i,l}] = \Theta_i(1 - \Theta_i).$$

En utilisant $\hat{\Theta}_i$ comme estimateur non biaisé de Θ_i, $\mathbb{V}[X_{i,l}]$ peut-être approchée par :

$$\mathbb{V}[X_{i,l}] \simeq \hat{\Theta}_i(1-\hat{\Theta}_i).$$

Ainsi, on a :

$$\begin{aligned}\mathbb{V}[\hat{\Theta}_i] &= \mathbb{V}[\frac{1}{n'_k}\sum_{l=1}^{n'_k} X_{i,l}] = \frac{1}{n'^2_k}\mathbb{V}[\sum_{l=1}^{n'_k} X_{i,l}],\\ &= \frac{1}{n'^2_k} n'_k \mathbb{V}[X_{i,l}] \simeq \frac{\hat{\Theta}_i(1-\hat{\Theta}_i)}{n'_k}.\end{aligned}$$

L'intervalle de confiance $IC_{95}(\Theta_i)$ a une taille $|IC_{95}(\Theta_i)|$ pouvant être estimée par :

$$\begin{aligned}|IC_{95}(\Theta_i)| &= 2\alpha_{95}\mathbb{V}[\hat{\Theta}_i] \simeq 2\alpha_{95}\frac{\hat{\Theta}_i(1-\hat{\Theta}_i)}{n'_k},\\ &\simeq 2\alpha_{95}\frac{K_i n'_k - K_i^2}{n'^3_k} \simeq 2\alpha_{95}\frac{K_i(n'_k-K_i)}{n'^3_k}.\end{aligned}$$

Soit $m_{IC,k}$ la valeur moyenne des tailles des intervalles de confiance pour la trajectoire T_k, définie par :

$$m_{IC,k} = \frac{\sum_{i=2\ldots N'_k+1}|IC_{95}(\Theta_i)|}{N'_k}.$$

$|IC_{95}(\Theta_i)|$ (et donc $m_{IC,k}$) est une fonction décroissante de N'_k puisque K_j est une fonction décroissante de N'_k. Soit δ un paramètre d'échelle permettant une comparaison pertinente de $m_{IC,k}$ et de N'_k. Le critère de décision défini par l'équilibre entre le nombre intérieur d'état N'_k et la valeur moyenne des intervalles de confiance $m_{IC,k}$ est alors obtenu en choisissant $\tilde{N}'_k$ minimisant $m_{IC,k} + \delta N'_k$:

$$\tilde{N}'_k = \arg\min_{N'_k}(m_{IC,k} + \delta N'_k). \tag{5.2}$$

Il reste à choisir δ. Soit une distribution donnée, les estimations asymptotiques des proportions $\hat{\Theta}_i, i = 2\ldots N'_k+1$ sont constantes. Ainsi, si $m_{IC,k}$ est une fonction décroissante de n'_k, alors le facteur d'échelle permettant que $m_{IC,k}$ et N'_k soient comparés doit également avoir une forme décroissante en fonction de n'_k. Une fonction décroissante $\delta(n'_k) = \frac{\beta}{n'_k}$ (fonction décroissante ayant la même forme que celle de $m_{IC,k}$) est utilisée.

Tout d'abord, une valeur $\hat{\delta}$ donnant des distributions, pour le choix de $\tilde{N}'_k$, compactes et distinctes pour les différentes classes de trajectoires est choisie. Celle-ci est obtenue en effectuant une analyse discriminante linéaire du premier ordre sur δ. Cette analyse permet en effet de maximiser les distances inter-classes et minimiser les distances intra-classes sur les choix de $\tilde{N}'_k$. Le "bon" nombre d'états $\tilde{N}_{C_i}$, un pour chaque classe C_i, est obtenu en calculant la moyenne des $\tilde{N}'_k$ pour les instances de la classe

(en ne considérant pas les trajectoires menant à des choix de $\tilde{N}'_k$ isolés par rapport au reste de la classe, voir figure 5.7).

Ensuite, comme le montre la figure 5.5, les intervalles de δ menant au "bon" $\tilde{N}'_{C_i}$, pour les trajectoires ou groupes de trajectoires issus d'une même classe de trajectoires (et ce pour différentes classes de trajectoires), sont utilisés. Une régression sur les bornes supérieures et inférieures de ces intervalles est obtenue en utilisant une méthode d'estimation à l'aide d'une fonction inverse de la taille des ensembles de données telle que :

$$\delta(n'_k) = \frac{\beta}{n'_k}.$$

La valeur β est alors trouvée en effectuant une régression par moindres carrés, *i.e.* en minimisant l'écart quadratique entre la régression sur $\delta(n'_k)$ et les bornes supérieures et inférieures des intervalles. Des résultats très satisfaisants sont obtenus comme illustré à la figure 5.5, validant expérimentalement l'hypothèse selon laquelle δ est une fonction décroissante en n'_k et plus spécifiquement une fonction inverse de n'_k.

Cette première fonction $\delta(n'_k)$ obtenue est ensuite utilisée pour choisir un nouveau nombre d'états $\tilde{N}'_k$ pour l'ensemble des trajectoires considérées (à l'aide de l'équation 5.2), menant à une nouvelle régression, et ainsi de suite jusqu'à ce que la fonction $\delta(n'_k)$ soit stable (*i.e.*, jusqu'à ce qu'il n'y ait plus de changement dans la séquence de $\tilde{N}'_k$ associée aux trajectoires). Cette méthode permet d'avoir, pour toutes les classes de trajectoires, une fonction $\delta(n'_k)$ pertinente. Empiriquement, une valeur $\beta = 0.0175$ a été trouvée, de sorte que $\delta(n'_k) = \frac{0.0175}{n'_k}$ est une fonction d'échelle efficace permettant un choix automatique du nombre d'états, pour toute trajectoire de toute classe. La Fig. 5.7 résume les opérations permettant le choix de δ, avec tout d'abord la sélection de $\hat{\delta}$, le choix des $\tilde{N}'_{C_i}$ et le processus itératif de régression jusqu'à obtention de δ_β.

Afin d'effectuer une comparaison pertinente entre trajectoires (*i.e.*, de comparer des MMCQ ayant le même nombre d'états), le choix d'un unique nombre d'états $\tilde{N}'$ pour l'ensemble des trajectoires traitées est nécessaire. Ainsi, en utilisant la fonction $\delta(n'_k)$ décrite précédemment, le nombre d'états $\tilde{N}'$ choisi dans $[B_{1,k}, B_{2,k}]$ est donné par

$$\tilde{N}' = \arg\min_{N'} \sum_k (m_{IC,k} + \delta N'), \tag{5.3}$$

en prenant en compte l'ensemble des trajectoires (Fig. 5.6).

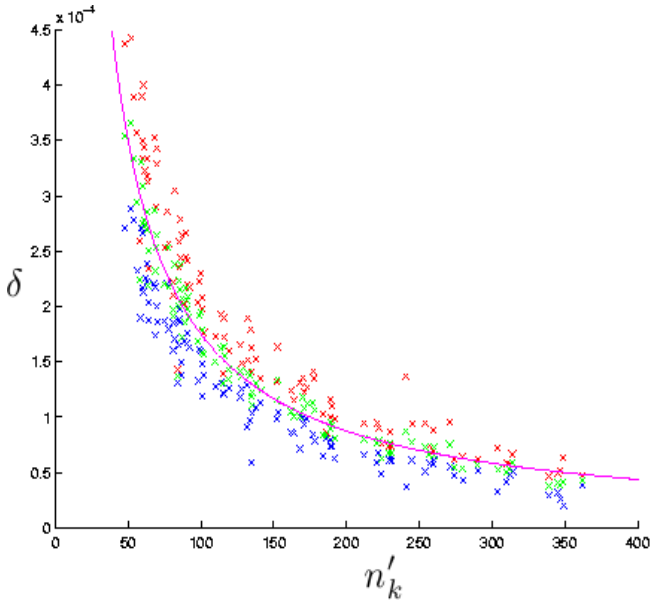

FIG. 5.5 – Intervalles de δ menant au "bon" N' choisi en fonction des tailles des ensembles de données considérés (trajectoires ou groupes de trajectoires issues de différentes classes de trajectoires). Les points en rouge et bleu correspondent respectivement aux bornes supérieures et inférieures de ces intervalles, les points verts étant leurs valeurs moyennes. La fonction en violet est la régression obtenue sur les points rouges et bleus en utilisant une méthode d'estimation par minimisation de l'erreur quadratique à l'aide d'une fonction inverse de la taille des données.

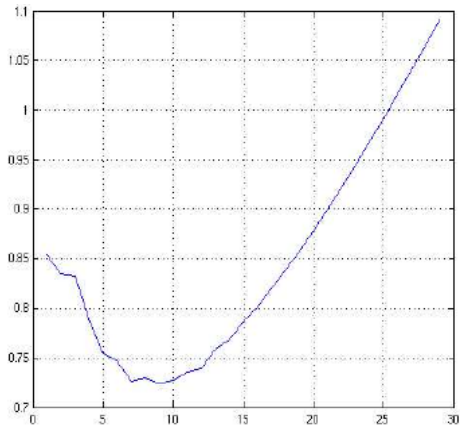

FIG. 5.6 – Fonction $\sum_k (m_{IC,k} + \delta N')$ utilisée pour choisir le nombre d'états intérieurs $\tilde{N}'$ pour les données des 10 classes de trajectoires de Formule1 (Fig. 6.3).

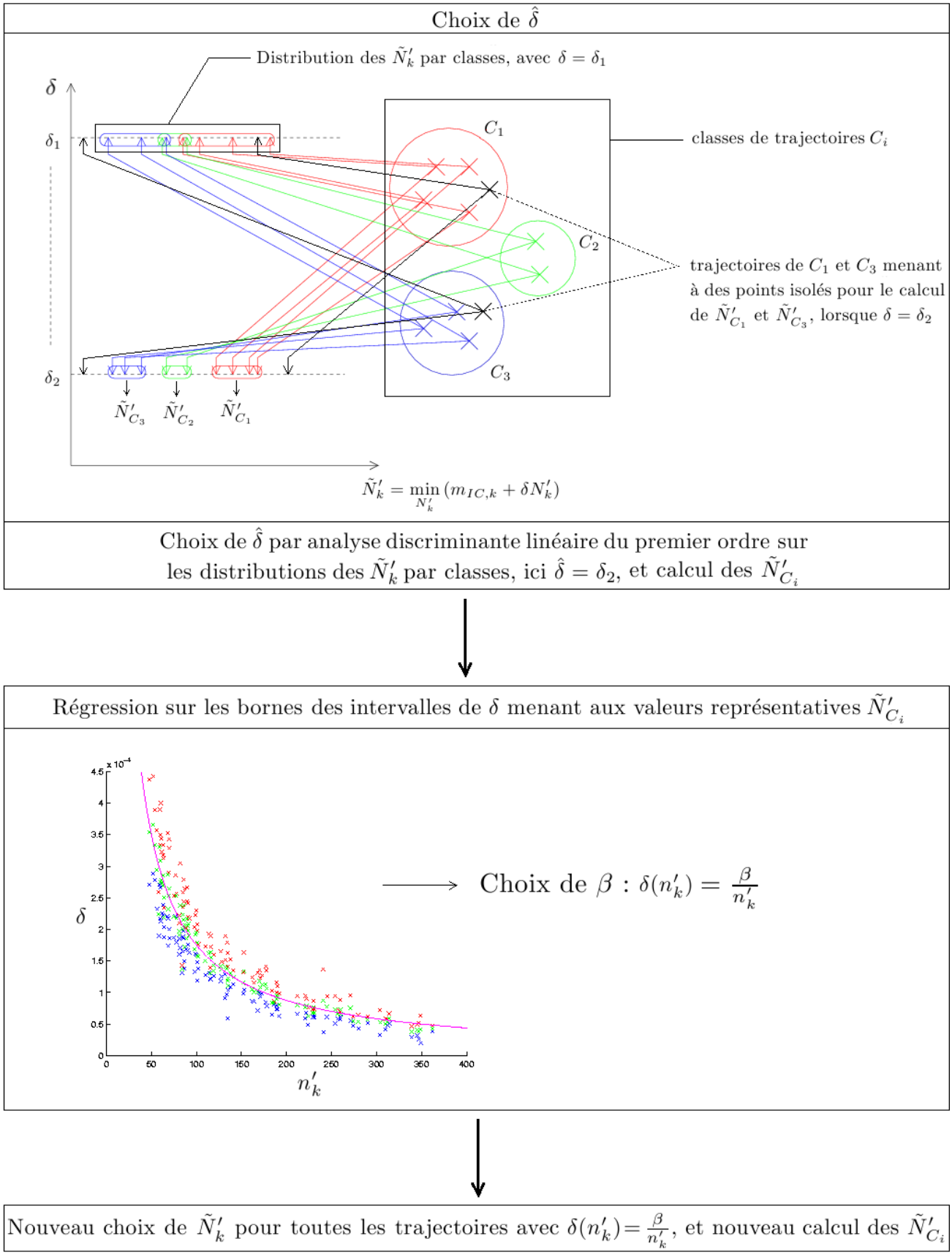

FIG. 5.7 – Schéma présentant l'algorithme de sélection de $\delta(n'_k)$.

5.1.3 Mesure de similarité entre trajectoires

Afin de comparer deux trajectoires représentées par des MMCQ, une mesure de similarité doit être définie. Une mesure de similarité a été proposée par Rabiner [Rabiner 89] pour comparer deux MMC. Étant donnés deux MMC ayant pour paramètres λ_1 et λ_2 ($\lambda_i = (A_i, b_i, \pi_i), i = 1, 2$), on a

$$D(\lambda_1,\ \lambda_2) = \frac{1}{n_2}[\log\ P(O^{(2)}|\lambda_2) - \log\ P(O^{(2)}|\lambda_1)], \tag{5.4}$$

où $O^{(j)} = \dot{\gamma}_1\dot{\gamma}_2...\dot{\gamma}_{n_j}$ est la séquence des mesures associées à la trajectoire T_j (de taille n_j) sachant λ_j, estimée à l'aide d'un algorithme de Viterbi, et $P(O^{(j)}|\lambda_i)$ exprime la probabilité d'observer $O^{(j)}$ avec le modèle λ_i.
La version symétrisée de cette mesure de similarité est

$$D_s(\lambda_1, \lambda_2) = \frac{1}{2}[D(\lambda_1, \lambda_2) + D(\lambda_2, \lambda_1)]. \tag{5.5}$$

Dans la méthode par MMCQ proposée, une modélisation spécifique est définie pour chacune des trajectoires T_k, associée à l'intervalle $[B_{1,k}, B_{2,k}]$. Ainsi, pour comparer deux trajectoires T_i et T_j, les ensembles de paramètres λ_i^j, λ_i^i, λ_j^j et λ_j^i sont introduits, où λ_i^j correspond aux paramètres trouvés pour T_j avec la modélisation associée à T_i (*i.e.*, en considérant les observations $\dot{\gamma}$ de T_j dans $[B_{1,i}, B_{2,i}]$). La distance (ou, plus exactement, mesure de similarité, qu'on appellera abusivement distance dans la suite) croisée symétrique D_c entre deux trajectoires T_i et T_j est définie par :

$$D_c(\lambda_i, \lambda_j) = \frac{1}{4}[D(\lambda_i^i, \lambda_j^i) + D(\lambda_j^i, \lambda_i^i) + D(\lambda_i^j, \lambda_j^j) + D(\lambda_j^j, \lambda_i^j)]. \tag{5.6}$$

Ainsi, la modélisation par MMCQ permet qu'une observation $\dot{\gamma}$ dans $[B_{1,k}, B_{2,k}]$ puisse appartenir à chacun des états (*i.e.*, $\forall i, P[\dot{\gamma}_t \mid q_t = S_i] \neq 0$, voir figure 5.3). Cela permet de traiter le cas des petites trajectoires, en empêchant d'avoir des valeurs nulles lors de l'estimation de A (voir équation 5.1). En effet, si les observations ne pouvaient être associées à tous les états définis (*i.e.*, $P[\dot{\gamma}_t \mid q_t = S_i] = 0$, pour certains i), alors la procédure de comptage utilisée (éq. 5.1) pour l'estimation des paramètres des MMCQ pourrait conduire, par manque d'observations, à des valeurs nulles dans la matrice A. Des valeurs nulles dans la matrice de transition des MMCQ correspondent à des transitions interdites et à des distances infinies entre trajectoires pouvant néanmoins être proches. Ainsi, la distance entre deux trajectoires T_k et T_l similaires, à une unique observation près pour T_l correspondant à une transition interdite par le MMCQ associé à T_k, serait alors infinie, ce qui est une situation non désirée que la modélisation proposée permet d'éviter.

5.2 Tâches de reconnaissance vidéo considérées

Cette section est dédiée aux tâches de reconnaissance vidéo qui ont été explorées à l'aide de la modélisation de trajectoires par MMCQ décrite précédemment. La distance

D_c (équation 5.6) entre MMCQ est utilisée pour la comparaison de trajectoires.

Elle est tout d'abord exploitée dans la méthode que nous avons définie pour la reconnaissance supervisée de plans vidéos à l'aide de trajectoires. Pour cette tâche, un ensemble de classes de trajectoires connues *a priori* est pris en compte. Chacune de ces classes est caractérisée par un ensemble de trajectoires d'entraînement. Les trajectoires tests, dont on veut connaître la classe d'appartenance, sont ensuite associées à l'une des classes de trajectoires.

Le clustering de plans vidéos à l'aide de trajectoires a également été envisagé. Au contraire de la classification, aucune connaissance *a priori* n'est utilisée dans cette tâche de reconnaissance de contenu. Les groupes de trajectoires sont formés au fur et à mesure de la procédure de regroupements de trajectoires. Les classes de trajectoires finalement obtenues sont choisies à l'aide d'un critère d'arrêt du processus de regroupement de trajectoires (section 2.2).

Enfin, une technique de détection de trajectoires anormales révélatrices d'évènements inattendus a été mise en place. Pour cette dernière tâche de reconnaissance de contenu dans des vidéos, des classes de trajectoires vidéos connues *a priori* sont considérées. Les trajectoires tests sont comparées à chacune de ces classes. La détection d'évènements rares est effectuée en calculant, pour chaque classe, un critère d'appartenance. Les plans vidéos traités correspondent alors à des plans inattendus si les trajectoires n'appartiennent à aucune des classes définies *a priori*.

5.2.1 Reconnaissance d'évènements à l'aide de trajectoires

Nous présentons tout d'abord les méthodes de reconnaissance, supervisée et non-supervisée, de trajectoires s'appuyant respectivement sur les techniques d'agrégation par liens moyens [Han 01] et de classification non-supervisée hiérarchique (section 2.2).

5.2.1.1 Reconnaissance supervisée d'évènements à l'aide de trajectoires

La reconnaissance supervisée d'évènements dans des vidéos à partir des trajectoires extraites s'appuie sur les modèles MMCQ que nous avons introduits. Cette tâche correspond donc à une requête par l'exemple. Chaque classe de trajectoires est représentée par l'ensemble des MMCQ correspondant aux trajectoires d'apprentissage de cette classe. Il s'agit de déterminer, pour une trajectoire test, la classe la plus proche. Cette classe sera alors la classe d'appartenance de la trajectoire test.

Nous avons recours, pour la phase de reconnaissance, à une technique d'agrégation par lien moyen. Nous calculons la moyenne des distances entre la trajectoire testée T_k et toutes les trajectoires T_l de la classe C_i par :

$$D_{lm}(T_k, C_i) = \frac{\sum_{T_l \in C_i} D_c(T_k, T_l)}{\#C_i}, \tag{5.7}$$

où D_{lm} est la distance par lien moyen (d'où la notation lm) entre une trajectoire et une classe de trajectoires. La trajectoire T_k est associée à la classe la plus proche, c'est-à-dire la classe C_i correspondant à la plus petite valeur de $D_{lm}(T_k, C_i)$.

5.2.1.2 Clustering d'évènements à l'aide de trajectoires

Un autre objectif peut être d'obtenir un clustering (ou classification non-supervisée) de plan vidéos à l'aide des trajectoires extraites. Aucune information *a priori*, pour cette tâche, n'est disponible. Le but est donc de créer des regroupements de trajectoires en paquets (ou classes, clusters) homogènes, chaque sous-ensemble de trajectoires partageant des caractéristiques communes, définies à l'aide de mesures de distance (ici, la distance D_s définie en section 5.1.3).

Nous avons exploité à cette fin une technique de clustering binaire hiérarchique ascendant. Cette technique a été choisie pour la stabilité des résultats (voir section 2.2. En effet, notre volonté a été de tester les propriétés de comparaison de trajectoires des la modélisation MMCQ, nous avons donc choisi une méthode hiérarchique dans le but d'obtenir les résultats de clustering les plus stables sans nous soucier de l'optimisation du temps de calcul. Chaque trajectoire représente initialement une classe. Les regroupement effectué par a méthode de clustering binaire hiérarchique ascendant utilisent la distance entre deux clusters C_i et C_j définie par :

$$D_{lm}(C_i, C_j) = \frac{\sum_{T_k \in C_i, T_l \in C_j} D_c(T_k, T_l)}{\#C_i \#C_j}. \tag{5.8}$$

Le critère d'arrêt considéré dans nos travaux est de donner *a priori* le nombre de clusters désirés, noté k_{clust} (voir section 2.2).

5.2.2 Détection d'évènements rares ou inattendus à l'aide de trajectoires

Détecter des évènements inattendus (ou, de façon équivalente, rares ou anormaux) peut s'avérer d'un intérêt important pour des tâches de vidéo-surveillance ou de traitement automatique de vidéos de sport. En effet, devant le nombre grandissant de caméras de vidéo-surveillance, la demande en moyen humain permettant l'analyse de tels contenus vidéos explose. La modélisation et la détection d'évènements doivent aider l'être humain dans la surveillance des situations observées. Un système d'alertes semi-automatisé peut, par exemple, être mis en place à l'aide de telles méthodes. Chacune de ces alertes nécessitant une confirmation humaine (pour éviter les fausses alarmes) avant la mise en place d'une intervention adaptée à la situation observée.

Cette tâche est à nouveau appréhendée à l'aide des MMCQ décrits précédemment. Tout d'abord, un ensemble de classes prédéfinies est utilisé. Chacune de ces classes est représentée par ses trajectoires d'apprentissage et les MMCQ correspondants. Pour chacune de ces classes de trajectoires C_i, la trajectoire la plus représentative T_{l_i} (*i.e.* la trajectoire ayant la distance moyenne avec les autres trajectoires de sa classe la plus

faible) est retenue. Nous considérons ensuite la distribution des distances des trajectoires de la classe C_i à T_{l_i}. Pour chaque classe C_i, nous calculons ainsi le maximum et la variance des distances intra-classes à T_{l_i}, respectivement notés R_i et σ_i.

Une trajectoire T_k est définie comme évènement inattendu si, pour toutes les classes C_i,

$$D_c(T_k, T_{l_i}) > R_i + \sigma_i. \tag{5.9}$$

5.2.3 Modélisation "globale" de trajectoires à l'aide de modèles de Markov cachés par quantification

Nous avons aussi développé des variantes de la méthode décrite pour le traitement de trajectoires vidéos. Plutôt que d'utiliser un intervalle (donc une quantification) par trajectoire, un seul intervalle $[B_1, B_2]$ est retenu pour l'ensemble des trajectoires traitées (*i.e.*, une quantification réalisée sur l'ensemble des $\dot{\gamma}$ observés pour l'ensemble des trajectoires). Cette méthode est désignée dans la suite par MMCQ globale.

La méthode MMCQ permet de comparer plus finement les distributions de $\dot{\gamma}$ associés à chaque trajectoire et de considérer les valeurs extrêmes des $\dot{\gamma}$ de chaque trajectoire comme une information importante correspondant à des phases restant significatives du mouvement d'un objet donné. Dans la suite, les méthodes développées pour les trois tâches utilisant les MMCQ sont exploitées avec les MMCQ globaux en exploitant la distance D_s à la place de la distance D_c.

5.3 Autres méthodes utilisées pour des fins de comparaison

À notre connaissance, il n'existe pas d'autres méthodes permettant de comparer véritablement une à une des trajectoires (potentiellement petites) respectant les propriétés d'invariance considérées. Ainsi, pour comparer de façon pertinente notre méthode à d'autres méthodes d'analyse de trajectoires vidéos, nous avons dû développer des techniques de traitement (supervisé et/ou non-supervisé) de trajectoires vidéos, ou du moins étendre des approches existantes pour les rendre comparables à notre méthode.

Les deux premières classes de méthodes classiques sont celles s'appuyant sur des histogrammes, la première inspirée des MMCQ global et la seconde des MMCQ. Une troisième classe largement explorée est celle exploitant des modélisations plus "classiques" de MMC où les distributions des observations correspondant à chaque état sont modélisées par des MMG. Une quatrième classe de méthodes très utilisée pour des tâches de classification, comprend les méthodes avec Séparateurs à Vastes Marges (SVM).

Les comparaisons avec les techniques d'histogrammes permettront de comprendre l'impact de la prise en compte explicite de la causalité temporelle. La comparaison

avec les MMC pourra mettre en valeur l'efficacité de la méthode par MMCQ proposée pour le traitement non-supervisé de petits ensembles d'observations, une des contraintes de l'utilisation effective des MMG étant de disposer d'un nombre suffisant d'observations. Les SVM sont des outils dont les nombreuses et récentes utilisation ont montré l'efficacité pour des tâches de reconnaissance supervisée. Ils permettront de mettre en valeur l'efficacité de la modélisations utilisant les MMCQ et notamment de la distance D_s.

Pour les trois classes de méthodes mises en place pour cette évaluation comparative, nous avons utilisé la distance entre trajectoires par lien moyens (voir sections 5.2.1.1 et 5.2.2). Le mode de choix du nombre d'états dans les MMCQ est aussi conservé (voir section 5.1.1). Les instanciations des méthodes relevant de ces trois classes, qui seront comparés à la méthode que nous avons proposée, sont décrites dans les sections suivantes.

5.3.1 Méthode par histogrammes globaux

Nous avons mis en œuvre la méthode suivante de comparaison d'histogrammes. Nous conservons le principe de la modélisation de trajectoires vidéos par MMCQ globaux (section précédente 5.2.3). Les histogrammes sont construits de manière analogue aux MMCQ globaux, avec une quantification similaire prise dans $[B_1, B_2]$ pour toutes les observations $\dot{\gamma}$ de toutes les trajectoires.

Nous avons recours à la distance de Bhatacharyya D_b entre deux histogrammes normalisés h_i et h_j d'observations $\dot{\gamma}_t$, correspondant respectivement aux trajectoires T_i et T_j. Elle est donnée par :

$$D_b(T_i, T_j) = \sqrt{1 - \sqrt{\sum_{q=1}^{N} h_q^i h_q^j}}, \tag{5.10}$$

où h_q^i est la valeur du bin q de l'histogramme associé à la trajectoire T_i.

5.3.2 Méthode par comparaison croisée d'histogrammes

Nous avons également développé une méthode de reconnaissance basée sur la comparaison croisée entre histogrammes. De manière similaire au traitement de trajectoires par MMCQ, un intervalle $[B_{1,k}, B_{2,k}]$ est associé à chaque trajectoire T_k, ainsi qu'une quantification permettant de définir un histogramme pour chaque trajectoire. La distance de Bhattacharyya entre histogrammes (voir section 5.3.1) est étendue à une distance croisée entre histogrammes. Nous entendons par là que chaque histogramme étant défini sur un intervalle différent, la distance D_{cb} entre deux trajectoires T_i et T_j

est cette fois définie par :

$$D_{cb}(T_i, T_j) = \sqrt{1 - \sqrt{\sum_{q_i=1}^{N_i} h_{q_i}^i h_{q_i}^j}} + \sqrt{1 - \sqrt{\sum_{q_j=1}^{N_j} h_{q_j}^i h_{q_j}^j}}, \tag{5.11}$$

où $h_{q_j}^i$ est la valeur du bin q_j (correspondant au bin q de la modélisation associée à la trajectoire T_j) lorsque la trajectoire T_i est traitée dans $[B_{1,j}, B_{2,j}]$.

5.3.3 Modèles de Markov cachés avec modélisation par mélanges de gaussiennes

Nous avons développé une méthode par MMC inspirée des travaux de Porikli [Porikli 04a]. La distance de Rabiner entre MMC, dont les états ainsi que les probabilités d'observation conditionnelles sont obtenues en utilisant des modèles de mélange de gaussiennes (MMG), a été étendue à l'analyse des $\dot{\gamma}_t$. Ainsi, un MMC continu "left-to-right" (voir section 1.3.1) est utilisé. Dans la modélisation considérée, une unique gaussienne (composant le MMG) modélise les probabilités d'observation conditionnelles à chaque état. L'initialisation des MMG est faite par un algorithme de type *k-means* (voir section 2.4). Les clusters ainsi obtenus permettent de calculer les paramètres (centres et variances) de chaque mode du mélange initial.

Pour déterminer le nombre d'états à utiliser dans le MMC, un critère de type ICL ("Bayesian Information Criteria", voir section 2.3) a été utilisé. Néanmoins, comme pour les méthodes d'estimation *BIC* et *AIC*, le critère *ICL* a tendance à surestimer le nombre d'états à utiliser. Ainsi, les meilleurs résultats ont été obtenus en utilisant la méthode *ICL* tout en limitant le nombre maximal G de gaussiennes utilisées par un MMC/MMG. La valeur de G est commune à l'ensemble des MMC/MMG et à été expérimentalement fixée à 3.

• Reconnaissance supervisée d'évènements à l'aide de classes de trajectoires par MMC/MMG

Pour la reconnaissance supervisée de trajectoires, un MMC/MMG est créé pour chaque classe de trajectoire *a priori* considérée. L'ensemble des observations $\dot{\gamma}$ des trajectoires associées à une classe d'évènements est donc utilisé pour le calcul des paramètres du mélange de gaussiennes à l'aide des algorithmes *k-means* (pour l'initialisation) et *EM* (voir section 2). La reconnaissance d'une trajectoire vidéo T_k est finalement réalisée en associant une trajectoire à une classe (parmi les classes C_i) par maximum de vraisemblance, *i.e.*, à la classe C_{max} telle que :

$$C_{max} = \arg\max_{C_i} P(T_k|C_i). \tag{5.12}$$

Cette méthode sera dans la suite notée méthode MMC/MMG (1).

• Clustering et reconnaissance supervisée d'évènements à l'aide de trajectoires par MMC/MMG

Pour le clustering non-supervisé de trajectoires, un MMC/MMG est créé pour chaque trajectoire. Le clustering et réalisé en utilisant la distance de Rabiner D_s entre MMC/MMG (voir section 5.1.3) à l'aide d'une méthode de lien moyen (voir 5.2.1.1), le clustering étant obtenu à l'aide d'une classification hiérarchique ascendante.

Cette distance D_s entre MMC/MMG modélisant une unique trajectoire, nécessaire pour le clustering, a également été utilisée pour la reconnaissance supervisée (ou classification) de trajectoires. Similairement à ce qui est fait pour les MMCQ, une technique d'agrégation par lien moyen est exploitée. La moyenne des distances entre la trajectoire testée T_k et toutes les trajectoires T_l d'une classe C_i par :

$$D_{lm}(T_k, C_i) = \frac{\sum_{T_l \in C_i} D_s(T_k, T_l)}{\#C_i}, \tag{5.13}$$

où D_{lm} est la distance par lien moyen entre une trajectoire et une classe de trajectoires. La trajectoire T_k est associée à la classe la plus proche, c'est-à-dire la classe C_i correspondant à la plus petite valeur de $D_{lm}(T_k, C_i)$.

Ces méthodes de clustering et de classification seront notées MMC/MMG (2).

5.3.4 Méthode avec Séparateur à vastes marges (SVM)

Un outil efficace de classification est le SVM (Séparateur à Vaste Marge) [Burges 98]. En entrée des SVM, nous avons choisi d'utiliser les paramètres des MMCQ globaux associés aux trajectoires (voir section 5.2.3). Les données en entrée des SVM doivent être représentées sous formes de vecteurs. Par conséquent, pour chaque trajectoire, nous avons créé un vecteur contenant les paramètres du MMCQ global correspondant. Par exemple, considérons le MMCQ global λ_i correspondant à la trajectoire T_i (pour des facilités de présentation, nous développons un exemple avec N = 3). Nous avons

$$A_i = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \ \pi_i = [a_1 \ a_2 \ a_3].$$

Alors $X_i = [a_{11} \ a_{12} \ a_{13} \ a_{21} \ a_{22} \ a_{23} \ a_{31} \ a_{32} \ a_{33} \ a_1 \ a_2 \ a_3]$ sera le vecteur caractérisant la trajectoire T_i. Nous utilisons une technique de classification par SVM à l'aide d'un noyau gaussien. Les résultats obtenus sont issus d'un schéma de classification "un contre tous", les paramètres du SVM ayant été fixés à l'aide d'une validation croisée. Enfin, la méthode de sélection du nombre d'états des MMCQ globaux est celle proposée en 5.1.2. La technique SVM développée sera utilisée dans la tâche de reconnaissance supervisée d'évènements à l'aide de trajectoires.

5.4 Conclusion

Dans ce chapitre, nous avons décrit la méthode originale que nous avons définie. Elle exprime explicitement les causalités temporelles des observations. Elle forme une variante des MMC discrets en mettant en avant une quantification adaptée des variables considérées (les descripteurs locaux de trajectoires $\dot{\gamma}$ exprimant forme et dynamique) et leurs propriétés causales. Elle est désignée par méthode MMCQ. Elle a l'avantage de pouvoir appréhender effectivement les trajectoires vidéos de longueur réduite. Nous avons développé une technique de nature statistique pour choisir automatiquement le nombre d'états de ces MMCQ. Cette méthode peut d'ailleurs s'étendre au choix du nombre de bins d'une quantification quelconque pour des tâches de classification. Nous avons montré comment cette méthode est exploitée pour aborder les tâches de reconnaissance supervisée d'évènements vidéos à l'aide de trajectoires, de clustering automatique de trajectoires, ainsi que la détection d'évènements inattendus. Un ensemble de méthodes ont aussi été décrites, relevant des classes d'approches par histogrammes, par MMC continus et par SVM. Cette évaluation expérimentale est précisément l'objet du chapitre suivant.

Chapitre 6

Applications et expérimentations

"Ce qui ne signifie pas, comme on le dit trop, que la vérité historique soit toujours et en tout insaisissable. Il en va de cette vérité comme de toutes les autres : on se trompe plus ou moins"

Marguerite Yourcenar - ***Mémoires d'Hadrien***

Nous décrivons, dans ce chapitre, les expérimentations ayant permis de mettre en application les méthodes développées dans le chapitre précédent. Nous abordons tout d'abord les tâches de reconnaissance d'évènements dans des vidéos de Formule1 et ski à l'aide de trajectoires extraites de plans vidéos. Puis, nous avons également exploité les méthodes MMCQ, ainsi que les méthodes développées à des fins de comparaison, pour la reconnaissance de formes. Les formes sont alors interprétées comme des trajectoires que l'on parcours à vitesse constante.

6.1 Reconnaissance d'évènements dans des vidéos à l'aide de trajectoires

Des trajectoires extraites de plans vidéos issus de programme TV de Formule1 sont utilisées pour tester les méthodes développées dans le chapitre précédent. La figure 6.1 présente des séquences d'images appartenant, pour chaque ligne, à un même plan vidéo dont est extraite une trajectoire. Ces trajectoires ont été extraites de ces plans vidéos à l'aide des méthodes de suivi et d'estimation du mouvement de la caméra exposées dans la section 4.1. Les différentes classes de trajectoires de Formule1 sont

rassemblées dans la partie inférieure de la figure 6.3. Une classe est composée de trajectoires issues de plans vidéos filmés par une même caméra. Les différentes classes correspondent ainsi aux différentes caméras placées le long du circuit, à des endroits stratégiques du circuit. On peut s'apercevoir que les trajectoires utilisées sont largement similaires aux trajectoires 3D des Formules1, à une homographie près, les mouvements des mobiles étant quasiment planaires. Il est important de préciser que les trajectoires exploitées ici sont de "faibles" tailles, composées en moyenne d'une centaine de points.

Des vidéos de programmes TV de ski ont également été exploitées. De la même manière que pour les vidéos de Formule1, des trajectoires de skieurs ont été calculées dans des plans vidéos issus de ces vidéos. La figure 6.2 présente des images des plans vidéos de descente et de slalom utilisées (une séquence d'image associées à un plan donné dans chaque ligne). Ces trajectoires ont été extraites des plans vidéos à l'aide des mêmes méthodes de suivi et d'estimation du mouvement de la caméra que pour les trajectoires de Formule1. Dans la partie supérieure de la figure 6.3, les classes de skieurs sont présentées. Chaque classe de trajectoires de skieurs, de manière identique aux classes de trajectoires de Formule1, correspondent à des trajectoires obtenues dans des plans vidéos extraits d'une même caméra. Deux classes de trajectoires ont été extraites d'un programme de compétitions de descente, trois classes de trajectoires ayant été extraites de vidéo filmant une compétition de slalom.

Les vidéos exploitées, tant pour les programmes de grand prix de Formule1 que pour ceux d'épreuves de ski, sont filmées par des caméras mobiles. Les caméras sont mobiles dans le sens où elles ont des mouvements de rotations permettant de suivre les mobiles observés (respectivement les Formules1 et les skieurs). Ces mouvements de rotation, ainsi que les effets de zoom sont pris en compte dans l'estimation du mouvement de la caméra (voir section 4.1) pour le recalage des trajectoires dans le plan image initial de la vidéo. Comme nous l'avons souligné dans le chapitre 4, les initialisations des procédures de suivi sont faites à la main.

FIG. 6.1 – Images de plans vidéos filmées par deux caméras différentes dans un programme TV de Formule1 à deux endroits donnés sur le circuit. Chaque ligne de la figure présente une séquence d'image associée à un plan donné. Les trajectoires extraites par une technique de suivi sont sur imprimées sur les images.

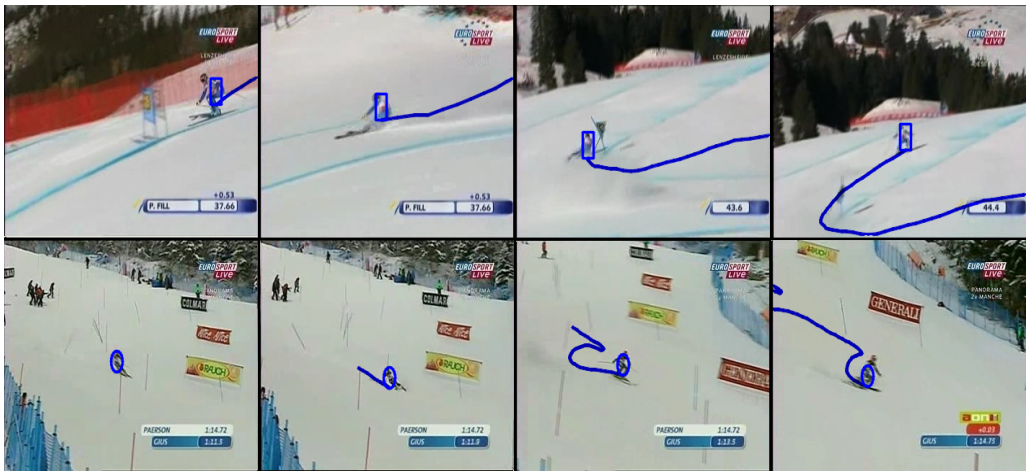

FIG. 6.2 – Images de deux plans vidéos filmés par deux caméras différentes dans un programme TV de ski (le premier extrait d'une descente et le deuxième d'un slalom). Chaque ligne de la figure présente une séquence d'image associée à un plan donné. Les trajectoires extraites par le suivi sont sur imprimées sur les images.

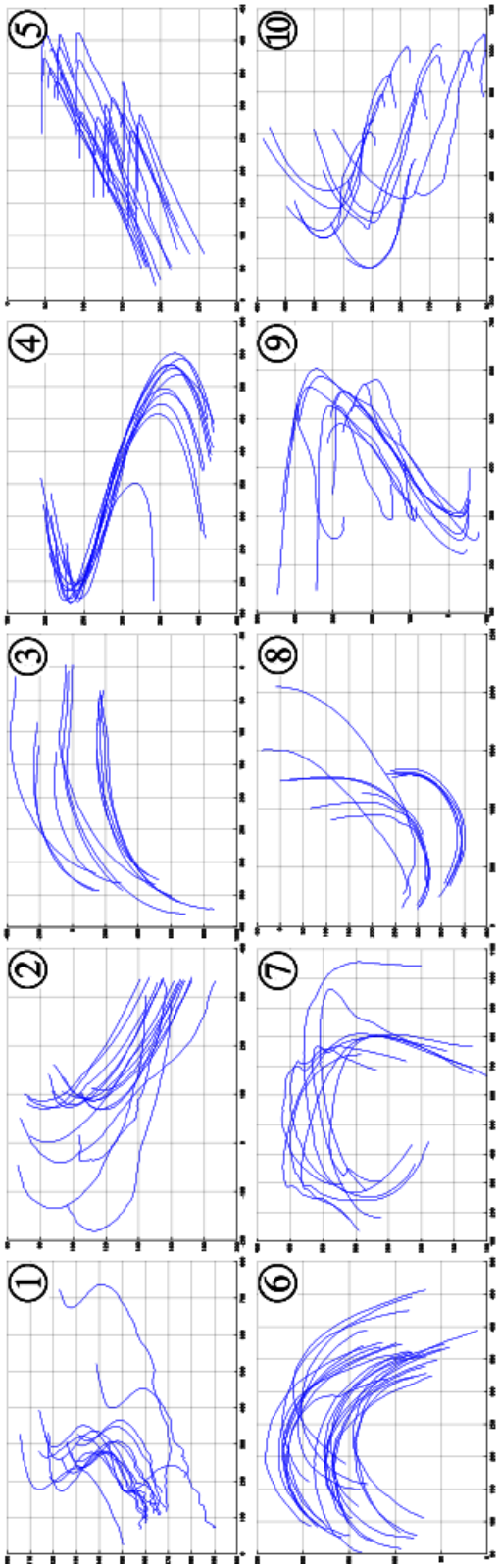

FIG. 6.3 – Tracé des 10 classes de trajectoires (149 trajectoires) extraites d'un programme TV de Formule1, classe par classe. Une classe est composée de trajectoires issues de plans vidéos filmés par une même caméra. Les différentes classes correspondent ainsi aux différentes caméras placées à des endroits stratégiques du circuit.

6.1.1 Reconnaissance supervisée d'évènements à l'aide de trajectoires

Nous décrivons maintenant les résultats de reconnaissance supervisée obtenues en traitant les classes de trajectoires de Formules 1 et de skieurs. La méthode de traitement des trajectoires par MMCQ (5.1.1) est comparée à la distance de Bhattacharyya croisée (5.3.2), la méthode MMC/MMG (5.3.3), la méthode MMCQ globaux (5.2.3), la distance de Bhattacharyya (5.3.1) ainsi qu'avec la méthode de classification par SVM (5.3.4).

Afin d'évaluer les performances de ces méthodes, une validation croisée "leave-one-out"a été mise en œuvre [Hastie01]. Le principe de cette méthode est le suivant. Soit un ensemble de classes de trajectoires comprenant n trajectoires dont on connaît la classe d'appartenance. Afin d'effectuer la classification d'une trajectoire donnée (trajectoire test), les $n-1$ trajectoires restantes sont utilisées afin d'entraîner les classes pour la reconnaissance. Cette opération est effectuée pour chacune des n trajectoires, les résultats de classification correspondent alors au pourcentage de trajectoire classée dans leur classe d'appartenance.

La méthode d'estimation *ICL* à été employée pour les paramètres de nombre d'états des MMC/MMG. Néanmoins, comme nous avons pu le souligner en section 5.3.3, la méthode *ICL* ayant tendance à surestimer le nombre d'états à retenir. Ainsi, les meilleurs résultats ont été obtenus en utilisant la méthode *ICL* tout en limitant le nombre maximal G de gaussiennes utilisées par un MMC/MMG (expérimentalement choisi égal à 3).

	Pourcentage de bonne classification			
# classes	4	6	8	10
MMCQ	100	99	96	92.7
Bhattacharyya croisée	98.2	98	95.2	92.7
MMCQ globale	96.4	99	94.4	91.9
Bhattacharyya	94.5	96	93.6	86.6
SVM	96.4	99	92.8	87.2
MMC/MMG (2)	98.2	98	95.2	89.9
MMC/MMG (1)	98.2	96	93.6	83.2

TAB. 6.1 – Comparaison des résultats de reconnaissance supervisée pour les trajectoires extraites de vidéos de Formule1 obtenus en utilisant une méthode de validation croisée "leave-one-out". Les groupes de 4, 6, 8 et 10 classes sont respectivement composés des classes 1 à 4, 1 à 6, 1 à 8 et 1 à 10 introduits dans la figure 6.3.

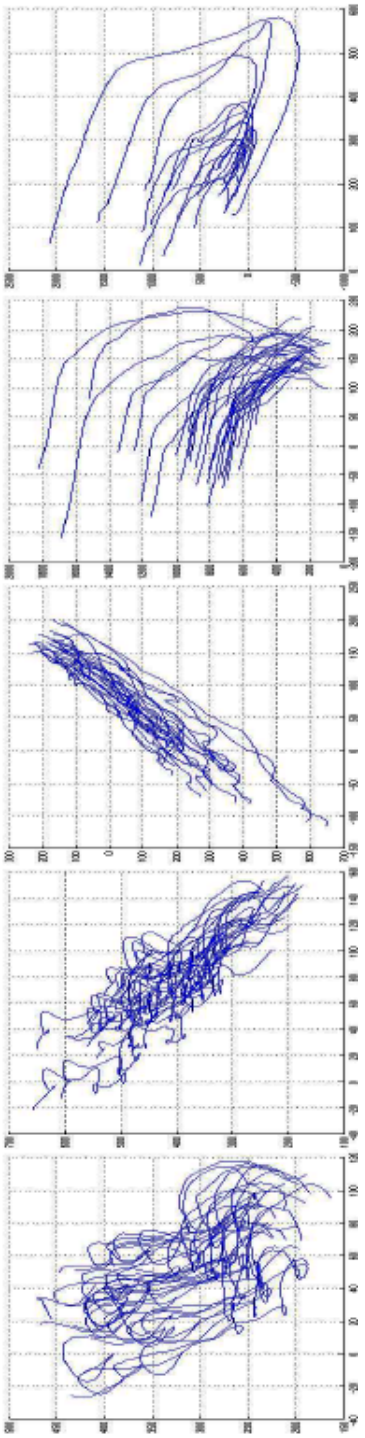

FIG. 6.4 – Tracé des 5 classes de trajectoires (134 trajectoires) extraites de programmes TV de compétitions de ski, classe par classe. Une classe est composée de trajectoires correspondant à des plans vidéos filmés par une même caméra. Les trois classes de gauche correspondent à des trajectoires issues de plans extraits d'une retransmission TV d'un slalom, les deux suivantes à des plans extraits d'une retransmission TV de descente.

Les tableaux 6.1 et 6.2 contiennent respectivement les résultats obtenus sur les ensembles de trajectoires vidéos de Formule1 (voir figure 6.3) et de ski (voir figure 6.4). Ainsi, les résultats de classification obtenus sur 4, 6, 8 et 10 classes par la méthode MMCQ sont respectivement de 100%, 99%, 96% et 92.7%. Les groupes de 4, 6 et 8 classes sont respectivement composés des classes 1 à 4, 1 à 6 et 1 à 8 introduits dans la partie inférieure de la figure 6.3. Cette évaluation sur des trajectoires issues de vidéos permet de mettre en valeur les propriétés de la méthode MMCQ, les résultats de classification les plus précis étant obtenus avec cette méthode. La comparaison avec la distance croisée de Bhattacharyya, qui offre également des résultats satisfaisants de classification, montre que la prise en compte de la causalité temporelle pour le traitement de trajectoires permet d'améliorer sensiblement la reconnaissance d'évènement dans des vidéos. La pertinence d'une modélisation de chaque trajectoire à l'aide d'un MMCQ "dédié" est mise en évidence par la comparaison avec la méthode MMCQ globale et la distance de Bhattacharyya. Ces deux dernières méthodes (ainsi que la méthode SVM) obtiennent des résultats moins précis. Par ailleurs, les résultats de classification obtenus avec ces méthodes de nature plus "globale" s'avèrent meilleurs si l'on introduit 6 classes plutôt que 4 classes. Cela montre le manque de stabilité de ces méthodes dû à la quantification globale effectuée. Les méthodes MMC/MMG ne parviennent pas à classer correctement les trajectoires comme peuvent le faire les autres méthodes évaluées. Cela souligne la flexibilité des modèles MMCQ par rapport à des modélisations plus complexes lorsque l'on veut traiter des ensembles de données de faibles tailles.

	Pourcentage de bonne classification
MMCQ	92.4
Bhattacharyya croisée	91.7
MMCQ globale	91.7
Bhattacharyya	91.7
SVM	91
MMC/MMG (2)	84.2
MMC/MMG (1)	78.2

TAB. 6.2 – Comparaison des résultats de reconnaissance supervisée pour les 5 classes de trajectoires extraites de vidéos de ski en utilisant la méthode de validation croisée "leave-one-out".

Un avantage important de la représentation invariante introduite par les descripteurs $\dot{\gamma}$ est d'être robuste aux erreurs possibles dans la compensation d'un mouvement de caméra. En effet, l'estimation du mouvement global dans l'image induit par celui de la caméra peut s'avérer imprécis dans certains cas, générant des effets résiduels sur les trajectoires calculées comme on peut le voir dans la figure 6.3. Toutes les classes

contiennent des trajectoires affectées d'erreurs de recalage, se traduisant par des décalages en translation, et plusieurs classes présentent des erreurs d'échelle (*e.g.*, les classes 4 et 8 de la partie inférieure de la figure 6.3, les deux classes de droite de la partie supérieure de la figure 6.3).

⋆ ***Classification de classes composées de trajectoires issues de caméras différentes***

Des classes de contenus vidéos différentes ont également été mise en place. Ainsi, nous avons réalisé des tests de classification en exploitant les classes de trajectoires de ski "slalom" et "descente". La classe "slalom" étant composée des trois classes de gauche de la partie supérieure de la figure 6.3, la classe "descente" des deux classes de droite. Des résultats parfaits ont été atteints pour la classe "descente" et la classe "slalom", cela pour toutes les méthodes testées (MMCQ, MMCQ global, Bhattacharyya, Bhattacharyya croisée, MMC/MMG (1) et MMC/MMG (2)).

Nous avons également construit des nouvelles classes de contenus vidéos à l'aide des trajectoires de Formule1. Les classes de trajectoires de Formule1 suivantes ont été utilisées : "virage" (classe 2 dans la partie inférieure de la figure 6.3), "virage serré" (regroupement des classes 5 et 9), "courbe légère" (classe 3), "chicane" (regroupement des classes 1, et 10) et "demi-cercle" (regroupement des classes 6, 7 et 8). Des classifications parfaites ont également été obtenues, à l'aide des 6 méthodes exploitées, en prenant en compte ces cinq classes.

Il est à noter que les méthodes MMC/MMG (1) et (2) sont, pour cette nouvelle tâche, implémentées avec une architecture ergodique (voir la définition en section 1.3.1). En effet, les classes sont composées de trajectoires extraites de vidéos filmées par des caméras différentes, et contiennent donc des trajectoires de tailles différentes. Or, afin que les méthodes MMC/MMG puissent traiter pertinemment des trajectoires de tailles différentes, les MMC/MMG doivent être avoir une topologie ergodique et non "left-to-right" (des explications supplémentaires peuvent être trouvés dans l'introduction du chapitre 5).

Ces résultats montrent l'intérêt de la représentation choisie qui, de par ses invariances et la prise en compte des informations de formes et de dynamique, permet l'analyse de mouvements issus de caméra mobiles différentes. En effet, quelles que soient les méthodes utilisées, la représentation à l'aide de $\dot{\gamma}$ a permis de réaliser des classifications parfaites pour les deux classes de contenu dans des vidéos de ski "slalom" et "descente", ainsi que pour les cinq classes de contenu de vidéo de Formule1 "virage", "virage serré", "courbe légère", "chicane" et "demi-cercle".

6.1.2 Clustering d'évènements à l'aide de trajectoires

Le clustering de trajectoires, à l'aide d'une technique d'agrégation par liens moyens appliquée à la méthode de clustering hiérarchique ascendant proposée dans le chapitre 5, a été testé parmi les dix classes de trajectoires de Formule1 présentées en partie inférieure de la figure 6.3. Pour cette tâche, aucune classe *a priori* n'est définie. L'ensemble des trajectoires est regroupé au fur et à mesure du processus de clustering hiérarchique ascendant (section 2.2) en paquets (ou clusters). Afin d'effectuer cette tâche, les différentes distances entre trajectoires proposées dans le chapitre 5 ont été exploitées.

Les taux de clustering correct ont été calculés en comparant les clusters obtenus avec les classes de trajectoires. En effet, même si aucune information sur les classes n'est utilisée par la méthode de clustering exploitée, nous connaissons les compositions des classes de trajectoires. Nous avons donc comparé les clusters formés avec les classes de trajectoires et nous avons ainsi pu calculer le taux de bon clustering. Le tableau 6.3 présente les résultats de bons clustering obtenus sur les quatre ensembles de trajectoires considérés.

	Pourcentage de bon clustering			
# classes	4	6	8	10
MMCQ globale	96.4	96	88.8	80
Bhattacharyya	83.6	90	80.8	71.8
MMCQ	89.1	79	73.6	65.8
Bhattacharyya croisée	100	92	85.6	71.8
MMC/MMG (2)	83.6	89	68	65.8

TAB. 6.3 – Comparaison des résultats de clustering pour les trajectoires extraites de vidéos de Formule1. Les groupes de 4, 6, 8 et 10 classes sont respectivement composés des classes 1 à 4, 1 à 6, 1 à 8 et 1 à 10 de la partie inférieure de la figure 6.3.

Les groupes correspondant aux 4, 6, 8 et 10 classes sont respectivement composés des classes 1 à 4, 1 à 6, 1 à 8 et 1 à 10 décrites dans la figure 6.1. Les meilleurs résultats de clustering ont été obtenus avec la méthode MMCQ global. Les résultats de clustering obtenus, sur les ensembles correspondant aux 4, 6, 8 et 10 classes, par la méthode MMCQ global sont respectivement de 96.4%, 96%, 88.8% et 80%.

Afin d'obtenir ces résultats, nous avons défini, pour chacun des quatre ensembles, un critère d'arrêt k_{clust} (voir section 5.2.1.2), *i.e.* le nombre de cluster devant être formés *a priori* par la méthode de classification hiérarchique ascendante. Pour l'ensemble composé de 4 classes, le critère d'arrêt ayant permis d'obtenir les résultats présentés

dans le tableau 6.3 est $k_{clust} = 4$. De même, pour les ensembles composés de 6, 8 et 10 classes, les critères d'arrêt correspondant sont $k_{clust} = 7$, $k_{clust} = 9$ et $k_{clust} = 11$.

Les résultats de clustering obtenus sur les classes de trajectoires de Formule1 montrent qu'en présence de séquences d'observations de faibles tailles, les méthodes basées sur une quantification des observations (*i.e.*, les méthodes Bhattacharyya, MMCQ global, Bhattacharyya croisée et MMCQ) sont supérieures aux modélisations par MMC-/MMG.

Contrairement aux résultats obtenus pour la reconnaissance supervisée de trajectoires vidéos, les méthodes modélisant séparément chaque trajectoire (*i.e.*, les méthodes Bhattacharyya croisée et MMCQ) obtiennent des résultats inférieurs à la méthode MMCQ global. Ces résultats montrent l'intérêt d'une prise en compte globale des observations lorsqu'aucune connaissance *a priori* n'est disponible sur les structures des classes de trajectoires, contrairement à la reconnaissance supervisée qui implique l'existence de classes prédéfinies.

⋆ ***Clustering sur des ensembles de trajectoires issues de caméras différentes***

Comme pour les tests réalisés pour la reconnaissance supervisée, des clusterings parfaits des trajectoires vidéos de skieurs en deux classes "slalom" et "descente" ont été obtenus, et ce quelle que soit la méthode de clustering utilisée (MMCQ, MMCQ global, Bhattacharyya, Bhattacharyya croisée, MMC/MMG (1) et MMC/MMG (2)).

De plus si l'on considère les ensembles de trajectoires de Formule1 "virage", "virage serré", "courbe légère", "chicane" et "demi-cercle", un taux de bon clustering de l'ordre de 90% a été atteint avec la méthode MMCQ global. Ce résultat, le meilleur parmi les méthodes testées, montre les propriétés de la méthode MMCQ global pour le clustering de trajectoires.

6.1.3 Détection d'évènements inattendus à l'aide de trajectoires

Des expérimentations pour la tâche de détection d'événements inattendus ont également été conduites avec la méthode de détection par MMCQ décrite en section 5.2.2. Comme cette méthode obtient les meilleurs résultats de reconnaissance supervisée (voir section 6.1.1), ainsi elle semble la mieux adaptée pour la détection supervisée d'évènements inattendus par rapport aux classes apprises de comportements "normaux".

Nous avons pu extraire, dans le vidéos de programmes TV de compétitions de Formule1 et de ski, cinq évènements inattendus correspondant aux classes présentées dans la figure 6.3, un dans les vidéos de ski et quatre dans les vidéos de Formule1. Un

évènement inattendu, dans la vidéo de ski, correspond à la chute d'un skieur. Quatre évènements inattendus ont donc également été extraits des vidéos de Formule1 : deux accidents, une sortie de route et l'apparition de la voiture de sécurité. Les trajectoires correspondantes ont été extraites et ont servi à tester les méthodes de détection d'évènements inattendus.

La figure 6.6 présente les trajectoires "normales" de Formule1 d'une classe, correspondant à la classe 1 dans la partie inférieure de la figure 6.3. Les trajectoires de trois évènements inattendus sont également présentées (évènement inattendu "apparition de la voiture de sécurité", "accident" et "sortie de route". Ces trajectoires d'évènements inattendus ont été extraites de plans vidéos filmés par la même caméra que les plans vidéos dont ont été extraits "normales" de la classe 1.

FIG. 6.5 – Images extraites de plans vidéos de Formule1. Chaque ligne correspond à un exemple d'événement inattendu ("accident", "apparition de la voiture de sécurité" et "sortie de route"). Chaque ligne de la figure présente une séquence d'image associée à un plan donné. Les trajectoires obtenues par le suivi sont sur-imprimées sur les images.

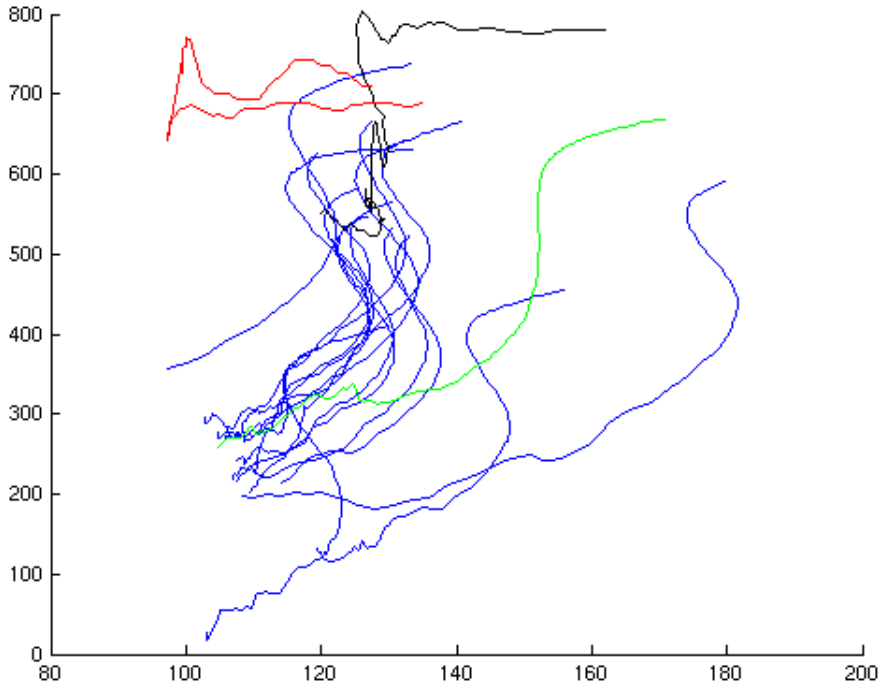

FIG. 6.6 – Tracé de trajectoires "normales" de Formule1 d'une classe en bleu (correspondant à la classe 1 dans la partie inférieure de la figure 6.3). Les trajectoires correspondant des trois évènements rares présentés dans la figure 6.5 sont tracés en vert (évènement inattendu "apparition de la voiture de sécurité"), en rouge (évènement inattendus "accident") et en noir (évènement inattendu "sortie de route"). Ces trajectoires ont été extraites de plans vidéos filmés par la même caméra que les plans vidéos dont ont été extraits les trajectoires "normales" présentées.

Les figures 6.5 et 6.7 présentent respectivement trois séquences d'images issues de vidéos de Formule1 et deux séquences d'images issues de vidéos de ski. Dans chaque cas, la première séquence d'image (première ligne) correspond à une vidéo au contenu "normal", *i.e.* dont le contenu correspondant à un événement attendu (la Formule1, le skieur suit le circuit normalement), alors que les autres séquences d'images (les autres lignes de la figure) correspondent à des événements inattendus parmi ceux décrits précédemment. Le critère défini dans la section 5.2.2 nous a permis de détecter ces événements comme inattendus.

Le tableau **??** contient, pour les quatre trajectoires associées aux événements inattendus observés dans les vidéos de Formule1, les valeurs du seuil $R_i + \sigma_i$ associé au critère utilisé (dans la deuxième ligne). Ils présentent également, dans la dernière ligne, les distances entre les trajectoires extraites des vidéos d'évènements inattendus

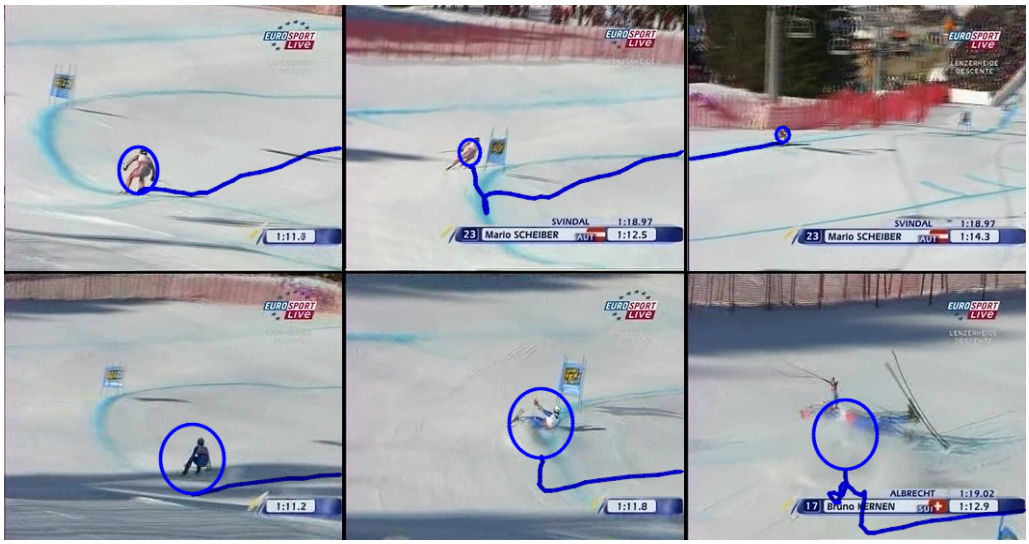

FIG. 6.7 – Images extraites de plans vidéos de ski filmés par une même vidéo. Chaque ligne de la figure présente une séquence d'image associée à un plan donné. Les trajectoires obtenues par le suivi sont sur-imprimées sur les images. La première ligne contient un exemple "normal" de la classe associée aux plans vidéos fournis par une même caméra. La seconde ligne présente un exemple d'événement inattendus (chute d'un skieur).

et les classes de trajectoires "normales" C_i (ici les classes 1 à 8 présentées dans la partie inférieure de la figure 6.3). Le tableau **??** présente ces mêmes données en considérant cette fois les dix classes de la partie inférieure de la figure 6.3. Enfin, le tableau **??** inclut, pour un événement inattendu dans une vidéo de ski, les valeurs du seuil $R_i + \sigma_i$, dans la deuxième ligne. Il présente également, dans la dernière ligne, les distances entre les trajectoires concernées et les cinq classes C_i (les classes contenues dans la partie supérieure de la figure 6.3).

Ainsi, la méthode MMCQ peut efficacement détecter des événements inattendus dans des vidéos. Concernant les trajectoires de Formule1 et en considérant huit classes de trajectoires vidéos (classes 1 à 8 de la partie inférieure de la figure 6.3), la méthode MMCQ a donné des résultats positifs de détection (voir tableau **??**) pour tous les événements inattendus observés dans les vidéos traitées. Il s'agit d'accidents ou de sorties de piste (ce qui se traduit par des différences de formes des trajectoires) ou de l'intervention de la voiture de sécurité (détectée par une différence de vitesse). La détection de l'apparition de la voiture de sécurité souligne la pertinence de la représentation choisie, à l'aide des descripteurs $\dot{\gamma}$ qui combine forme (courbure) et dynamique (vitesse de parcours). La voiture de sécurité suit bien entendu le circuit et a donc une

trajectoire de forme sensiblement similaire à celles des Formules 1 (la trajectoire correspondante est présenté dans la figure 6.6). Par contre, sa dynamique n'est pas la même puisque cette voiture n'a pas les mêmes caractéristiques de vitesse que les Formules 1. Le contenu est ainsi correctement détecté comme évènement inattendu.

Lorsque l'on considère, plutôt que huit classes de trajectoires de Formule1, l'ensemble des dix classes de trajectoires de Formule1 de la partie inférieure de la figure 6.3 (voir tableau **??**), les évènements inattendus autres que l'apparition de la voiture de sécurité sont toujours bien détectés. Néanmoins, l'apparition de la voiture de sécurité n'est plus détectée. Cet évènement est alors détecté comme appartenant aux classes 9 ou 10 de la partie inférieure de la figure 6.3, classes qui ont une forme proche de la trajectoire de la voiture de sécurité, avec des dynamiques qui rendent la détection de cet évènement inattendu difficile.

De la même manière, pour les trajectoires issues de vidéos de ski (tableau **??**), la méthode a correctement détecté, pour le cas des 5 classes de trajectoires de skieurs, la trajectoire associée à l'évènement "chute d'un skieur".

Ces résultats semblent montrer la pertinence de la méthode de détection proposée en section 5.2.2, ainsi que la pertinence de la représentation par les descripteurs $\dot{\gamma}$. Bien que les résultats soient intéressants, la détection de la voiture de sécurité n'a été effectuée avec huit classes, mais n'a pu être effectuée en considérant les dix classes de trajectoires. Une solution permettant la détection de cet évènement rare nécessiterait la prise en compte des informations visuelle de l'objet suivi, la voiture de sécurité devant être distinguable, visuellement, des formules 1.

De plus, dans les programmes TV exploités afin de créer les différentes classes de trajectoires, cinq évènements rares (d'où cette appellation) seulement ont été filmés par les caméras associées aux classes de trajectoires utilisées. Ainsi, afin de valider la méthode proposée pour la reconnaissance d'évènement rare, il serait indispensable d'effectuer des expérimentations supplémentaires sur d'autres plans issus de vidéos de Formule 1, de ski, ou plans issus d'autres types de contenu vidéo.

En effet, nous avons seulement exploité les classes pour lesquelles un nombre raisonnable de trajectoires ont été extraites (au moins une douzaine de trajectoires). Certaines caméras sont utilisées de façon épisodique et peu de plans vidéos sont alors observés. De plus, les trajectoires de ces classes devaient être efficacement recalées dans le plan image initial. Par exemple, pour de nombreuses caméras en plan rapproché des Formule1 et des skieurs, le calcul du mouvement de fond est inefficace (du fait du manque de points d'intérêts du fond observé) et les trajectoires correspondantes sont alors sans intérêt.

Classe	1	2	3	4	5	Statut
Seuil $R_i + \sigma_i$	1315	428	392	1442	454	
Chute d'un skieur	1723	1224	715	6086	3258	**détecté**

TAB. 6.4 – Les seuils de détection sont fournis dans la deuxième ligne, pour les classes de trajectoires considérées dans la détection d'événements inattendus (les 5 classes de la partie supérieure de la figure 6.3). La dernière ligne contient les distances entre la trajectoire représentant l'événement "accident" et les classes de trajectoires "normales".

Classe	1	2	3	4	5	6	7	8	Statut
Seuil $R_i + \sigma_i$	13	141	1189	69	55	2077	290	1167	
Accident 1	31	1823	3056	467	303	4667	1285	2313	**détecté**
Sortie de route	199	3487	3388	1260	64	3918	883	1544	**détecté**
Accident 2	70	1000	1788	1260	83	2885	373	1381	**détecté**
Voiture de sécurité	117	1069	1385	942	353	2673	5367	1362	**détecté**

TAB. 6.5 – Les seuils de détection sont fournis dans la deuxième ligne, pour 8 classes (apprises) de trajectoires de Formule1 considérées dans la détection d'événements inattendus (les classes 1 à 8 de la partie inférieure de la figure 6.3). Les lignes suivantes contiennent les distances entre la trajectoire correspondant à un évènement inattendu et les classes de trajectoires "normales". Les trajectoires correspondant aux événements "Accident 1", "Sortie de route" et "Voiture de sécurité" ont été filmées par la caméra associée à la classe 1, alors que l'événement "Accident 2" correspond à la classe 2.

Classe	1	2	3	4	5	6	7	8	9	10	Statut
Seuil $R_i + \sigma_i$	13	141	1189	69	55	2077	290	1167	467	1025	
Accident 1	31	1823	3056	467	303	4667	1285	2313	2470	1947	**détecté**
Sortie de route	199	3487	3388	1260	64	3918	883	1544	821	1278	**détecté**
Accident 2	70	1000	1788	1260	83	2885	373	1381	494	1612	**détecté**
Voiture de sécurité	117	1069	1385	942	353	2673	5367	1362	321	642	**non détecté**

TAB. 6.6 – Les seuils de détection sont fournis dans la deuxième ligne, pour les 10 classes (apprises) de trajectoires de Formule1 considérées dans la détection d'événements inattendus (les 10 classes de la partie inférieure de la figure 6.3). La lignes suivantes contiennent les distances entre la trajectoire représentant les évènements inattendus et les classes de trajectoires "normales". Les trajectoires correspondant aux événements "Accident 1", "Sortie de route" et "Voiture de sécurité" ont été filmées par la caméra associée à la classe 1, alors que l'événement "Accident 2" correspond à la classe 2.

6.2 Temps de calcul

Les temps de calcul entre trajectoires à l'aide des méthodes MMCQ et MMCQ globale sont faibles. Par exemple, le temps moyen pour le calcul de la distance D_c entre deux trajectoires de cent coordonnées chacune, avec un PC standard, est de l'ordre de 0.1 seconde. Ce temps de calcul comprend le calcul du nombre d'état nécessaire, des paramètres de chacun des deux MMCQ ainsi que leur comparaison. Les méthodes Bhattacharyya et Bhattacharyya croisée ont des temps de calcul du même ordre. Ces temps de calcul permettent d'envisager des applications en temps réel des comparaisons de trajectoires issues de vidéos.

Les méthodes MMC/MMG sont elles plus exigeantes. En effet, l'algorithme *EM* nécessite des temps de calcul plus importants que l'algorithme développé pour les MMCQ. Pour exemple, le temps moyen de calcul d'une distance D_s entre deux trajectoires de cent coordonnées chacune est de l'ordre de 0.5 seconde.

6.2.1 Extension des méthodes de comparaison de trajectoires à la reconnaissance de formes

La reconnaissance d'objets dans des images, à partir de leurs formes de contours, est un sujet de recherche important pour lequel de nombreux travaux ont été menés [Srivastava 03, Srivastava 07, Sebastian 01, Sebastian 04, Latecki 00, Bober 01]. Elle peut comporter des analogies avec la reconnaissance de trajectoires dans des vidéos considérées comme des courbes spatio-temporelles.

Nous cherchons à tester la pertinence de la caractérisation invariante des trajectoires (à l'aide du descripteur $\dot{\gamma}$, voir chapitre 2), ainsi que des méthodes proposées dans le chapitre précédent pour la reconnaissance de trajectoires. Pour cela, nous avons élaboré dans un premier temps des classes de formes bien définies.

Ces formes, observées dans des images, ont été extraites à partir d'une base de données disponibles en ligne [bdforme]. La figure 6.9 présente les images utilisées, images d'objets en noir et blanc, et les formes extraites à l'aide d'une technique simple de segmentation d'image. La segmentation a été réalisée par un algorithme basé sur un filtrage de Sobel. Les formes exploitées présentent des variations en termes de translation, de rotation et de facteur d'échelle, permettant ainsi de tester les propriétés d'invariance de la représentation par les descripteurs $\dot{\gamma}$.

Les formes d'objets ont donc été appréhendées comme des trajectoires, en parcourant la suite de points de leurs silhouettes à vitesse constante. Une forme approchée continue est créée à l'aide de l'approximation par noyaux décrite en section 4.2.1. Le nombre de points caractérisant les silhouettes résulte d'un échantillonnage uniforme

sur l'approximation continue des points de la forme. La figure 6.8 présente un échantillonnage en deux cents points obtenu sur une forme de chameau.

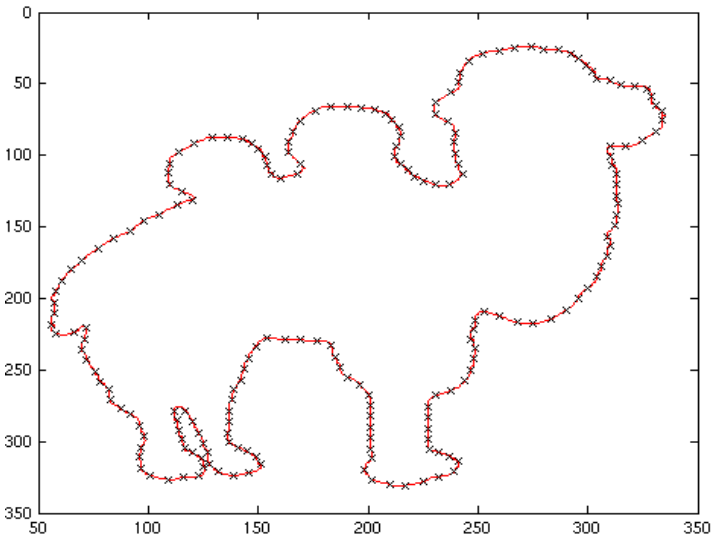

FIG. 6.8 – Échantillonnage en deux cents points (croix noires) effectué sur une forme de chameau (en rouge).

Les tests réalisés et décrits ci-dessous ici ont été menés en considérant dix classes de formes d'objets (pour un total de 190 formes), comprenant les formes de tasses, de voitures, de chameaux, de marteaux, de cloches, de chauve-souris, d'hélicoptères, de cafards, de garçons et de pommes (voir figure 6.9).

6.2.1.1 Reconnaissance supervisée de formes

Les résultats présentés dans le tableau 6.7 ont été obtenus en exploitant une méthode de validation croisée "leave-one-out". Le choix des paramètres de lissage h et de nombre d'états N'_k (pour les méthodes concernées) a été déterminé par les méthodes d'estimation développées respectivement aux chapitres 3 et 4.

Afin d'évaluer les performances de ces méthodes, la validation croisée "leave-one-out" exploitée en section 6.1.1 a été mise en œuvre. De même, la méthode *ICL* a été exploitée, tout en limitant le nombre maximal G de gaussiennes utilisées par un MMC/MMG.

Les résultats présentés correspondent aux meilleurs résultats obtenus parmi un certain nombre d'échantillonnages testés, en l'occurrence avec un échantillonnage de deux cents points.

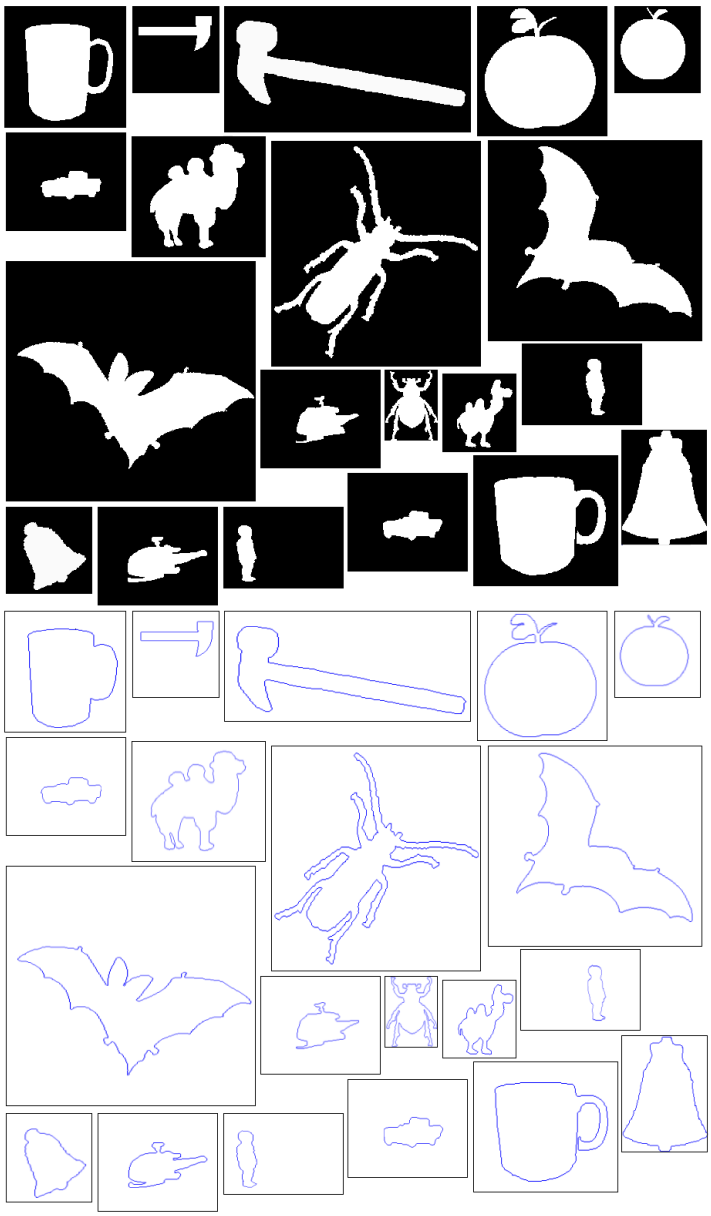

FIG. 6.9 – Partie supérieure : images en noir et blanc des objets traités. Partie inférieure : courbes de formes extraites des images de la partie supérieure et exploitées comme trajectoires supposées parcourues à vitesse constante.

Les résultats de reconnaissance, avec les différentes méthodes exploitées, sont indiqués dans le tableau 6.7. Tout d'abord, une moyenne de reconnaissance atteignant 97% a pu être obtenu. Ce résultat met en valeur la pertinence de la représentation à l'aide des descripteurs $\dot{\gamma}$, et notamment de ses propriétés d'invariance. En effet, les formes traitées contiennent des différences importantes en termes de translation, de rotation et d'échelle (voir figure 6.9), les bons résultats de classification confirment les

	Pourcentage de bonne classification
MMCQ	95.8
Bhattacharyya croisée	97
MMCQ globale	89.5
Bhattacharyya	90
SVM	76.8
MMC/MMG (2)	95.3
MMC/MMG (1)	77.9

TAB. 6.7 – Évaluation comparative des résultats de reconnaissance pour les dix classes de formes traitées, avec les différentes méthodes.

propriétés d'invariance, dans le plan image, des descripteurs $\dot{\gamma}$.

Les meilleurs résultats de reconnaissance ont été obtenus avec la distance croisée de Bhattacharyya entre histogrammes 5.3.2. Ces résultats, comparés à ceux de la distance de Bhattacharyya entre histogrammes 5.3.1, soulignent l'importance de traiter chaque trajectoire de façon indépendante. En effet, la distance croisée de Bhattacharyya entre histogrammes repose sur une quantification spécifique à chaque sous-ensemble de données (ici, à chaque forme), de manière équivalence à la méthode par MMCQ (5.1.1). La distance de Bhattacharyya, tout comme la méthode par MMCQ globale (5.2.3), a recours à une même quantification pour l'ensemble des données (*i.e.*, l'ensemble des formes).

De manière peu surprenante, la causalité temporelle semble ne pas être un élément déterminant pour la reconnaissance de formes puisque les méthodes MMCQ et MMC/MMG atteignent des performances inférieures à la distance de Bhattacharyya entre histogrammes. La comparaison entre la distance de Bhattacharyya et la méthode MMCQ globale confirme cette observation.

En effet, les méthodes MMCQ et MMC/MMG ont été construites pour prendre en compte la causalité temporelle intrinsèque aux trajectoires vidéos (ce qui n'est pas le cas des formes dans des images), les trajectoires évoluant dans un espace de trois dimensions (les deux dimensions de l'image ainsi que la dimension temporelle associée aux séquences d'images). Les trajectoires sont elles véritablement caractérisées par un ensemble ordonné de valeurs des descripteurs $\dot{\gamma}$. On touche là à la limite de l'analogie entre silhouettes d'objets et trajectoires.

Il est à noter que les MMC/MMG ont un nombre d'observations assez important (200 points dans ces expérimentations) pour une estimation fiable des paramètres des

MMG. Cela ne sera pas le cas pour les trajectoires de Formule1 et de skieurs traitées dans la suite de ce chapitre. Les méthodes MMCQ fournissent des résultats intéressants pour de longues séquences d'observations, en comparaison des méthodes MMC/MMG (1) et MMC/MMG (2). La méthode MMC/MMG (1) qui utilise un ensemble de trajectoires d'apprentissage pour chaque classe, obtient des résultats largement inférieurs à la méthode MMC/MMG (2). Cette dernière compare les trajectoires une à une avant d'effectuer la classification par une méthode de lien moyen.

6.2.1.2 Clustering de formes

Les méthodes exploitées pour le clustering de formes sont identiques à celles proposées pour le clustering d'évènements à l'aide de trajectoires extraites dans des vidéos de Formule1 (voir section). Ainsi, le clustering de formes, à l'aide d'une technique d'agrégation par liens moyens appliquée à la méthode de clustering hiérarchique ascendant a été testé parmi les dix classes de formes. Afin d'effectuer cette tâche, les différentes distances entre trajectoires proposées dans le chapitre 5 ont été exploitées, les silhouettes des formes étant appréhendées comme des trajectoires.

Les taux de clustering correct ont été obtenus en comparant les clusters formés avec les classes de formes. Aucune information sur les classes n'est utilisée mais nous connaissons les compositions des classes de formes. Ainsi, nous avons comparé les clusters produits avec les classes de formes et alors calculé le taux de bon clustering. Les résultats de bon clustering sont reportés dans le tableau 6.8. Le critère d'arrêt ayant permis d'obtenir les résultats présentés dans le tableau 6.8 est $k_{clust} = 12$.

	Pourcentage de bon clustering
MMCQ	79
Bhattacharyya croisée	72.1
MMCQ globale	64.7
Bhattacharyya	60.5
MMC/MMG (2)	66.8

TAB. 6.8 – Comparaison des résultats de reconnaissance pour les dix classes de formes (voir figure 6.9).

Le meilleur résultat de clustering sur les dix classes de formes est de 79%. Il a été obtenu avec la méthode MMCQ. Au contraire des résultats de reconnaissance supervisée de formes, la méthode Bhattacharyya croisée obtient un taux de clustering correct inférieur à ceux obtenus par la méthode MMCQ. Cela semble montrer que la prise en compte de la causalité temporelle peut aider le clustering de formes. Cette derniére observation est confirmée par les résultats obtenus par la méthode MMCQ globale

avec la méthode Bhattacharyya, la première obtenant un taux de clustering de 64.7% contre 60.5% pour la seconde.

L'importance d'une modélisation "dédiée" de chaque trajectoire est mise en évidence par la comparaison respective des méthodes MMCQ et Bhattacharyya croisée avec les méthodes MMCQ globale et Bhattacharyya. Tout comme pour la tâche de reconnaissance supervisée de formes, la méthode de clustering utilisant les MMC/MMG obtient les moins bons résultats de clustering.

6.3 Conclusion

Un ensemble de tests comparatifs portant sur des vidéos extraites de programmes TV de sport ainsi que sur des trajectoires représentant des formes d'objets a été conduit. Ces tests ont mis en valeur la pertinence de la représentation des trajectoires introduite et l'efficacité de la méthode MMCQ proposée pour différentes tâches de traitement de vidéos. Les méthodes proposées obtiennent de meilleurs résultats que les méthodes de reconnaissance par MMC "standards" (MMC/MMG), et ce en présence de trajectoires de tailles variables (petites pour les trajectoires vidéos, et plus grandes pour les formes d'objets). La méthode de classification par Bhattacharyya croisée montre de bons résultats de classification lorsque les causalités temporelles n'entrent pas en jeu. Pour des trajectoires vidéos dans lesquelles la causalité temporelle est à prendre en compte, les méthodes MMCQ et MMCQ global, permettent d'obtenir de meilleurs résultats de reconnaissance que celles reposant sur les distances de Bhattacharyya, les MMC/MMG ou les SVM. La méthode MMCQ apparaît adaptée aux tâches de classification, alors que la méthode MMCQ global est plus appropriée pour des tâches de reconnaissance non-supervisée (ou clustering).

Conclusion

La partie II a traité de l'analyse de trajectoires isolées. Nous avons tout d'abord spécifié une représentation de trajectoires vidéos permettant d'appréhender des trajectoires vidéos acquises par des caméras mobiles. Nous avons ensuite décrit une modélisation originale permettant de traiter des ensembles de données de faibles tailles, tout en tenant compte des causalités temporelles inhérentes aux trajectoires.

La représentation proposée, invariante, dans le plan image, aux transformations de translation, d'échelle et de rotation, permet d'utiliser des trajectoires issues de vidéos acquises par des caméras mobiles. De plus, la caractérisation des trajectoires définie exprime leurs propriétés de dynamique, au sens des variations de vitesse, que de forme, au sens des variations de courbure. Une représentation continue étant nécessaire pour la caractérisation des trajectoires, une méthode de régression a également été introduite avec sélection du paramètre de lissage.

Nous avons développé une modélisation originale, appelée modélisation par MMCQ pour modèles de Markov cachés avec quantification. Plus précisément, nous avons élaborer une modélisation par mélange de gaussiennes reposant sur une quantification uniforme des données, permettant l'exploitation d'ensembles de données de faibles tailles. La sélection du nombre d'états de la quantification est également prise en compte. Ainsi, l'ensemble de la méthode définie, de la représentation des trajectoires à la modélisation statistique qui en est faite, est automatique, fixation des paramètres comprise. Une distance entre trajectoires, reposant sur les MMCQ et inspirée d'une distance existante entre MMC, a également été développée.

Nous avons finalement testé l'ensemble de la méthode développée sur des tâches de classification, de clustering et de détection d'évènements inattendus. Plusieurs autres méthodes de classification et de clustering ont été spécifiées pour une évaluation comparative de la méthode proposée. Les expérimentations effectuées ont permis de montrer les atouts et l'efficacité de notre méthode MMCQ. Enfin, les résultats de détection d'évènements inattendus obtenus à l'aide de cette même méthode sont satisfaisants, montrant la flexibilité de notre approche et aussi la pertinence de la distance utilisée. Néanmoins, des tests sur d'autres ensembles de trajectoires (notamment pour la détection d'évènements rares) seront nécessaires afin de valider les méthodes proposée

Troisième partie

Reconnaissance de contenu vidéo par l'analyse des interactions entre trajectoires

Introduction

Dans cette partie, nous décrivons les méthodes développées pour l'analyse simultanée de plusieurs trajectoires observées dans des vidéos. Il s'agit de prendre en compte l'information mutuelle entre trajectoires que l'on désigne par les interactions entre trajectoires.

Nous allons considérer la segmentation temporelle de vidéos de sport en phase de jeu avec reconnaissance de ces dernières. Ainsi, les trajectoires des joueurs de squash et de joueurs de handball, dans le plan du terrain de jeu, sont exploitées pour la reconnaissance des différentes phases de jeu observées. Les trajectoires des joueurs de handball sont reconstruites à l'aide de deux caméras filmant les actions du dessus.

Pour chacune des tâches traitées, nous explorons des représentations permettant de prendre en compte les interactions entre les trajectoires observées. De plus, nous aurons recours à des modélisations markoviennes adaptées aux représentations choisies. Des expérimentations sur des données issues d'une base de données disponible en ligne seront menées.

Ainsi, cette première partie est structurée de la façon suivante :

- le **chapitre 7** présente l'aspect théorique des techniques d'analyse des interactions entre trajectoires. Les représentations correspondant aux tâches de reconnaissance de contenus dans des vidéos de sport sont proposées, ainsi que les modélisations markoviennes utilisées.

- dans le **chapitre 8**, nous testons les méthodes proposées sur deux applications, la segmentation temporelle et la reconnaissance des phases de jeu dans des vidéos de handball et de squash.

Dans la suite, la segmentation temporelle et la reconnaissance des phases de jeu dans des vidéos de sport sont effectuées de façon simultanée. Dans la suite de ce document, ces deux tâches seront englobées dans les notations **reconnaissance de phases de jeu** ou **interprétation** de vidéos de sport.

Chapitre 7

Reconnaissance de contenus vidéos à l'aide des interactions entre trajectoires

---

"[...] la complexité ne comprend pas seulement des quantités d'unités et interactions qui défient nos possibilités de calcul ; elle comprend aussi des incertitudes, des indéterminations, des phénomènes aléatoires. La complexité dans un sens a toujours affaire avec le hasard. [...] La complexité est donc liée à un certain mélange d'ordre et de désordre, mélange intime, à la différence de l'ordre/désordre statistique, où l'ordre (pauvre et statique) règne au niveau des grandes populations et le désordre (pauvre, parce que pure indétermination) règne au niveau des unités élémentaires. Quand la cybernétique a reconnu la complexité, ce fut pour la contourner, la mettre entre parenthèses, mais sans la nier : c'est le principe de la boîte noire ; on considère les entrées (inputs) dans le système et les sorties (outputs), ce qui permet d'étudier les résultats du fonctionnement d'un système, l'alimentation dont il a besoin, de relationner inputs et outputs, sans entrer toutefois dans le mystère de la boîte noire.
Or le problème théorique de la complexité, c'est celui de la possibilité d'entrer dans les boîtes noires."

Edgar Morin - ***Introduction à la pensée complexe***

Ce chapitre aborde l'exploitation des interactions entre trajectoires d'objets mobiles dans des vidéos dans le contexte de tâches de reconnaissance de contenus dy-

namiques. Les chapitres précédents portaient sur le traitement de trajectoires individuelles. Ce chapitre inclut la prise en compte de plusieurs trajectoires de façon simultanée et donc de leurs interactions. Nous exploiterons plus particulièrement les interactions entre joueurs dans des vidéos de sport. Plus précisément, il s'agira de la reconnaissance de phases de jeu à travers la segmentation temporelle automatique de ces vidéos de sport.

La tâche de segmentation et de reconnaissance simultanée de phase de jeu dans des vidéos de sport sera nommée **reconnaissance de phases de jeu** ou **interprétation**, dans la suite de cette partie.

7.1 Reconnaissance de phases de sports par interactions entre trajectoires

La compréhension des comportements et activités "complexes" observées dans des vidéos est d'une importance croissante dans le monde de la vision par ordinateur (voir section 3.2.3). Elle se trouve motivée par des applications dans des domaines variés tels que la vidéo surveillance, l'exploitation de vidéos de sport, la vidéo sur demande... Dans ces différents contextes, les trajectoires d'objets mobiles fournies par des méthodes de suivi peuvent être exploitées, fournissant une information de haut niveau pour la description des contenus dynamiques des vidéos.

Une approche classique pour l'analyse du contenu de vidéos repose sur un découpage préalable de vidéo en plans, puis, d'une caractérisation de ces plans par classification d'images "clés" [Gunsel 98, Kokaram 06, Denman03]. Ces méthodes trouvent leurs limitations notamment dans l'exploitation de documents vidéos qui ne comportent pas de montage, comme en vidéo-surveillance ou dans des vidéos de sport qui ne relèvent pas de diffusion télévisuelle. L'information délivrée par la détection et le suivi d'objets peut alors s'avérer d'un intérêt crucial pour le traitement de telles vidéos.

Différents travaux ont tenté d'adopter l'étude des objets mobiles pour l'analyse sémantique de vidéos. Günsel et al. [Gunsel 99] ont proposé une indexation de vidéos, filmée par une unique caméra, basée sur l'analyse des "objets vidéos". Ces objets vidéos sont définis comme les entités mobiles initialement détectées puis suivies dans une vidéo. Ils prennent en compte le mouvement des caméras ainsi que les trajectoires des objets vidéos et leurs interactions.

Comme évoqué dans le chapitre 3 l'état de l'art, des méthodes ont déjà été développées pour l'analyse "complexe" de vidéos [Oliver 00, Hongeng 03b, Natarajan 07a, Natarajan 07b], qui considèrent les interactions entre objets vidéos. La faible quantité

d'exemples exhibés montre la difficulté de définir des méthodes générales dans des cas divers et variés de vidéos et les limitations de telles approches. Ainsi, dans la suite de cette partie, nous nous proposons d'utiliser les trajectoires associées à des objets vidéos pour l'analyse de vidéos dans un contexte particulier, le contexte des vidéos de sport. En effet, les règles associées à ces activités ainsi que leur déroulement dans un espace défini *a priori* permettent de mettre à profit un cadre offrant un terrain d'expérimentations privilégié pour le développement de méthodes d'analyse de haut niveau de vidéos.

Nous proposons tout d'abord une méthode d'analyse de vidéos de squash filmées par une unique caméra selon une vue de dessus. Elle s'appuie sur les trajectoires des joueurs et sur leurs interactions. Les représentations de ces trajectoires sont construites invariantes aux transformations usuelles de rotation et de facteur d'échelle. Ainsi, les méthodes proposées pourront s'appliquer de façon directe à toutes trajectoires vidéos de squash filmées du dessus, quelles que soient les orientations ou les distances au terrain de la caméra. Nous avons aussi traité des trajectoires issues de vidéos de handball, également filmées du dessus, une caméra étant située au dessus du centre de chacune des deux moitiés du terrain. Les trajectoires sont reconstruites. L'objectif est d'élaborer une analyse des interactions entre les différents joueurs d'une même équipe.

Nous décrivons dans la section suivante la méthode que nous avons conçue pour l'analyse de trajectoires dans des vidéos de squash, avec une représentation invariante. La dernière section se focalisera sur le traitement de trajectoires reconstruites dans le plan du terrain de handball à l'aide de deux vidéos filmées du dessus.

7.1.1 Modélisation du jeu de squash par modèles semi-markoviens à l'aide des interactions entre trajectoires des joueurs

Nous souhaitons traiter des vidéos de squash pouvant provenir de configurations différentes de caméras mais restant placées au-dessus du centre de l'aire de jeu, sans avoir besoin de recaler précisément les trajectoires dans le terrain de squash. En conséquence, la modélisation de l'activité des joueurs de squash doit être invariante aux transformations de rotations et de facteur d'échelle, ce qui permettra qu'elle ne dépende pas de l'orientation de la caméra et de sa distance à l'action.

A cette fin, plusieurs représentations ont été testées. Nous considérons d'abord notamment la caractérisations invariante de trajectoires développée dans la partie précédente (partie II) puis la distance entre les joueurs.

7.1.1.1 Approximation continue des trajectoires de chaque joueur et calcul des descripteurs

Un objet vidéo, qui correspondra à un joueur de squash dans la suite de cette section, est représenté par une trajectoire vidéo notée T_k ($k = 1$ ou 2) correspondant aux positions successives de l'objet dans le plan image : $T_k = \{(x_{1,k}, y_{1,k}), .., (x_{n_k,k}, y_{n_k,k})\}$.

De la même manière qu'en section 4.2.1, nous appliquons une approximation par noyau de la trajectoire T_k afin de calculer efficacement les dérivées temporelles successives des coordonnées des trajectoires (*i.e.*, $\dot{u}_{t,k}$, $\dot{v}_{t,k}$, $\ddot{u}_{t,k}$ et $\ddot{v}_{t,k}$, u et v étant définis ci-dessous). Ainsi, une représentation continue de la trajectoire T_k est donnée par $\{(u_{t,k}, v_{t,k})\}_{t\in[1;n_k]}$ où :

$$u_{t,k} = \frac{\sum_{j=1}^{n_k} e^{-(\frac{t-j}{h})^2} x_{j,k}}{\sum_{j=1}^{n_k} e^{-(\frac{t-j}{h})^2}},\ v_{t,k} = \frac{\sum_{j=1}^{n_k} e^{-(\frac{t-j}{h})^2} y_{j,k}}{\sum_{j=1}^{n_k} e^{-(\frac{t-j}{h})^2}}.$$

L'approximation obtenue permet essentiellement de calculer les dérivées temporelles successives des coordonnées des trajectoires peu bruitées dans ce cas. Les trajectoires des joueurs de squash sont ici assez précises et l'utilisation d'un paramètre h minimal est suffisante (*i.e.*, $h = 1$).

Nous construisons une représentation invariante, pour chacun des joueurs, en exploitant la dérivée temporelle $\dot{\gamma}_t$ des orientations locales $\gamma_t = \arctan(\dot{v}_t/\dot{u}_t)$. Elle s'écrit

$$\dot{\gamma}_t = \frac{\ddot{v}_t\dot{u}_t - \ddot{u}_t\dot{v}_t}{\dot{u}_t^2 + \dot{v}_t^2} = \kappa_t.\|V_t\|,$$

avec $\kappa_t = \frac{\ddot{v}_t\dot{u}_t - \ddot{u}_t\dot{v}_t}{(\dot{u}_t^2+\dot{v}_t^2)^{\frac{3}{2}}}$ la courbure locale de la trajectoire et $\|V_t\| = (\dot{u}_t^2 + \dot{v}_t^2)^{\frac{1}{2}}$ l'amplitude de la vitesse locale au point (u_t, v_t).

Le vecteur représentant la dynamique d'un joueur de squash sera donc défini par les valeurs successives des variables $\dot{\gamma}$ au cours du temps, *i.e.*,

$$V_k = [\dot{\gamma}_{1,k}, \dot{\gamma}_{2,k}, ..., \dot{\gamma}_{n-1,k}, \dot{\gamma}_{n,k}].$$

7.1.1.2 Caractérisation de l'interaction entre les joueurs

La prise en compte des interactions entre objets vidéos (ici, entre joueurs) est, comme on le verra dans la section des résultats, d'une importance cruciale pour espérer une représentation pertinente d'activités complexes dans des vidéos. Nous représentons les interactions entre joueurs de squash par la distance. Ainsi, à chaque instant t, la distance entre les deux objets vidéos, correspondant aux deux joueurs de squash, et représentés par leurs trajectoires T_1 et T_2 est définie par (figures 7.1 et 7.2) :

$$d_t = \sqrt{(u_{t,1} - u_{t,2})^2 + (v_{t,1} - v_{t,2})^2}.$$

Plus spécifiquement, nous considérons la distance normalisée, *i.e.* :

$$\tilde{d}_t = \frac{d_t}{d_{norm}},$$

où d_{norm} est une valeur de référence, pouvant correspondre à la distance entre deux points connus du court de squash, ou, dans le cas où une telle valeur ne peut être calculée, la valeur moyenne observée des distances entre les deux joueurs.

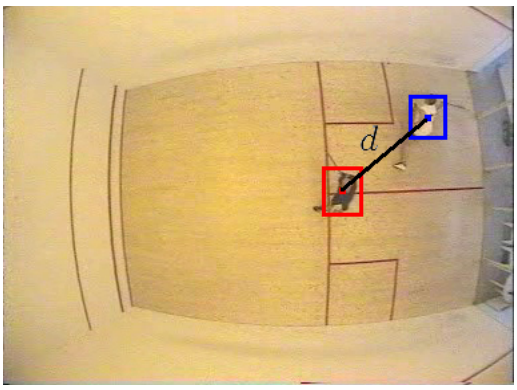

FIG. 7.1 – Une image tirée d'une vidéo de squash, les positions des deux joueurs correspondent aux rectangles rouge et bleu, d représente la distance entre les deux joueurs.

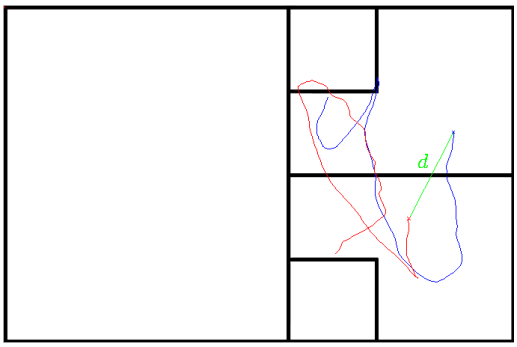

FIG. 7.2 – Représentation d'une portion de trajectoires pour les deux joueurs de squash (en bleu et en rouge). La distance observée entre les deux joueurs pour un instant donné est également tracée (en vert).

La distance $\tilde{d}_i$ est trivialement invariante aux transformations de translation ainsi que de rotation dans le plan image. La normalisation par d_{norm} permet, elle, d'avoir

une représentation invariante au facteur d'échelle (*i.e.*, à la distance entre la caméra et les objets observés). Le vecteur représentant les interactions entre joueurs de squash, pour chaque phase de jeu, sera donc composé des valeurs successives (une pour chaque image) de $\tilde{d}_i$:

$$D = [\tilde{d}_1, \tilde{d}_2, ..., \tilde{d}_{n-1}, \tilde{d}_n].$$

Ainsi, en utilisant d'une part les vecteur invariants V_k et d'autre part le vecteur D pour la représentation des activités dans une vidéo de squash, tous invariants aux transformations de rotation ainsi que de facteur d'échelle dans le plan image, on obtient bien une représentation invariante à ces mêmes transformations, tout en prenant en compte d'une part des mouvements particuliers de chaque joueur et d'autre part de leurs interactions.

Enfin, pour des raisons de temps de calcul, une procédure de regroupement des données "par paquet" a été mise en place. Pour les trois vecteurs de représentations considérés (*i.e.*, D, V_1 et V_2), des groupes de k_{group} données consécutives ont été formés. Pour chacun des groupes exhibés, la valeur moyenne est calculée afin de former de nouveaux vecteurs de représentation $\tilde{D}$, $\tilde{V_1}$ et $\tilde{V_2}$ (de taille k_{group} fois inférieure).

7.1.1.3 Modélisation des phases du jeu de squash par modèles cachés semi-markoviens

Nous avons recours aux modèles cachés semi-markoviens (MCSM) pour modéliser les activités observées dans les vidéos à partir des représentations choisies (*i.e.*, $\dot{\gamma}$ et $\tilde{d}$). Nous avons construit un schéma à deux niveaux. Les descripteurs des trajectoires $\dot{\gamma}$ et $\tilde{d}$ sont appréhendés à l'aide des MMC/MMG Pa. Ces modélisations par MMC/MMG Pa sont ensuite associées aux états de modèles cachés semi-markoviens (MCSM). Nous avons défini des MCSM hiérarchiques parallèles, modélisations proches de celles décrites récemment par Natarajan et al. [Natarajan 07a] pour la reconnaissance de langage des signes.

La couche de niveau supérieur de cette modélisation est composée des états des MCSM, appelés S'_i. La couche de niveau inférieur est concernée par la modélisation, à l'aide de MMC/MMG Pa, des descripteurs des trajectoires $\dot{\gamma}$ et $\tilde{d}$. La figure 7.3 propose une illustration de la modélisation hiérarchique mise en place.

⋆ *Modélisation des activités par MCSM*

Les états S'_i du MCSM correspondent aux phases d'activité. Les phases d'activité pour le jeu de squash sont les phases "jeu" et "non-jeu", respectivement notées par S'_1 et S'_2. Les MCSM permettent de modéliser les temps de séjour sd_i associés aux états

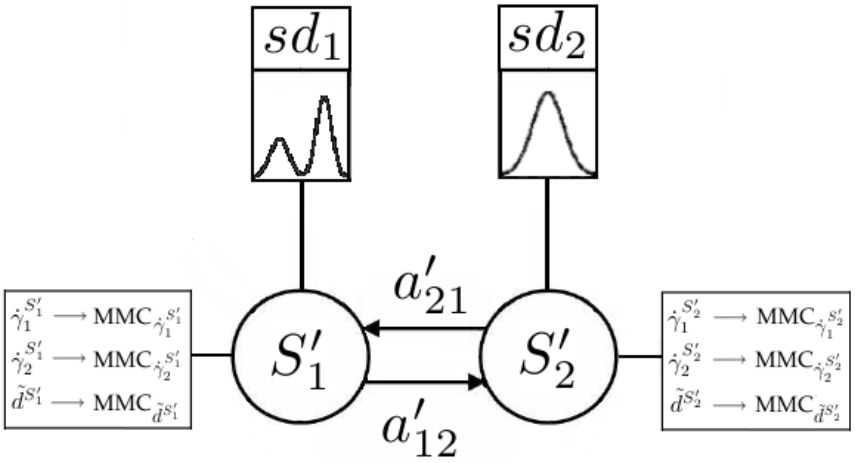

FIG. 7.3 – Modélisation par MCSM hiérarchique du jeu de squash, avec 2 états de niveau supérieur (S'_1 et S'_2) correspondant aux deux phases d'activité considérées ("jeu" et "non-jeu"). Chacune de ces phases d'activité est représentée par un MMC Pa à trois composantes, une pour chaque descripteur ($\dot{\gamma}_1^{S'_i}$, $\dot{\gamma}_2^{S'_i}$ et $\tilde{d}^{S'_i}$, les composantes du MMC/MMG Pa étant notées $MMC_{\dot{\gamma}_1^{S'_i}}$, $MMC_{\dot{\gamma}_2^{S'_i}}$ et $MMC_{\tilde{d}^{S'_i}}$), ainsi que par un MMG modélisant les temps de séjour (sd_i est le temps de séjour associé à la phase d'activité S'_i).

S'_i. Ces temps de séjour dans les états de niveau supérieur S'_i sont modélisés à l'aide de MMG.

Les paramètres des MMG sont calculés en utilisant des procédures de type "forward-backward" (voir section 2.3), les initialisations de ces procédures étant réalisées à l'aide d'algorithme de type "k-means" (voir section 2.4). L'ensemble des paramètres associé à la modélisation des temps de séjour sd_i par MMG sera noté ψ.

⋆ *Modélisation des représentations à l'aide de MMC Pa*

Les descripteurs de trajectoires et de leurs interactions $\dot{\gamma}_1$, $\dot{\gamma}_2$ et $\tilde{d}$ caractérisent chaque état de niveau supérieur S_i. Dans la figure 7.3, ils sont ainsi notés $\dot{\gamma}_1^{S'_i}$, $\dot{\gamma}_2^{S'_i}$ et $\tilde{d}^{S'_i}$. La modélisation de ces descripteurs, pour chaque état de niveau supérieur, est effectuée par MMC/MMG Pa (voir section 7.1.2.2). La figure 7.3 illustre la modélisation décrite.

Le critère d'estimation *ICL* (voir section 2.3) est exploité pour déterminer les nombre d'états des composantes MMC/MMG des MMC/MMC Pa. Il est à noter que le critère *ICL* exploité a tendance à surrestimer le nombre d'états à utiliser. Ainsi, nous avons

exploité la méthode *ICL* tout en limitant le nombre maximal G de gaussiennes utilisées par un MMC/MMG. La valeur de G est commune à l'ensemble des composantes MMC/MMG de tous les MMC/MMG Pa et à été expérimentalement fixée à 5.

⋆ Entraînement et algorithme de Viterbi pour la reconnaissance de phases de jeu dans des vidéos de squash

Ainsi, étant donné une séquence d'états S' contenant R segments, c'est-à-dire R phases d'activités successives, chacune associée à un unique état S'_i. Soit q_r l'index temporel du temps de fin du r^{me} segment, les observations du r^{me} segments sont $y_{(q_{r-1}+1,q_r]} = y_{q_{r-1}+1}, \ldots, y_{q_r}$ telles que $S'_{q_{r-1}+1} = \ldots = S'_{q_r}$. S' correspond à la séquence d'états de niveau supérieur des MCSM. A' est alors définie comme la matrice de probabilité de transitions entre états de niveau supérieur des MCSM aux temps $\{q_i\}$, calculée à l'aide des transitions entre phases d'activité observées dans la vidéo d'apprentissage. Nous adoptons une modélisation avec deux états de niveau supérieur (figure 7.3), et nous avons donc $a'_{21} = a'_{12} = 1$ et $a'_{11} = a'_{22} = 0$.

Les vidéos d'entraînement et, donc, les trajectoires qui en sont extraites, sont également utilisées afin de déterminer l'ensemble des paramètres des MMC/MMG Pa. L'ensemble des paramètres, pour toutes les composantes de tous les MM/MMG Pa est appelé ϕ. Il est composé des paramètres des MMC/MMG (B, A et π) pour chacune des trois composantes des MMC/MMG Pa, et ce pour chaque état de niveau supérieur S'_i.

L'ensemble des paramètres est donc défini par $\theta = \{A', \phi, \psi\}$ et est estimé par apprentissage supervisé. Ces paramètres sont utilisés dans la reconnaissance des phases d'activité temporelles. Nous pouvons ainsi réaliser une reconnaissance des phases de jeu S'_i dans les vidéos de squash à l'aide d'un algorithme de décodage de Viterbi pour MCSM (voir section 1.3.6).

L'algorithme de Viterbi est utilisé pour trouver la séquence d'états de niveau supérieur $\hat{S}$ des MCSM maximisant la log-vraisemblance,*i.e.*, telle que $\hat{S}' = \arg\max_{S'} \log P(y, S'|\theta)$. La vraisemblance $P(y, S'|\theta)$ est définie, pour une séquence d'observation y, et la séquence d'états de MCSM correspondante S', par :

$$
\begin{aligned}
P(y, S'|\theta) &= \prod_{r=1}^{R} P(S'_r|S'_{r-1}) \\
&\times \prod_{r=1}^{R} P(sd_i = q_r - q_{r-1}|\psi; S'_{q_r}) \\
&\times \prod_{r=1}^{R} P(y_{(q_{r-1},q_r]}|\phi; S'_{q_r}).
\end{aligned}
$$

Des modèles MCSM hiérarchiques parallèles ont été décrits dans cette section. Pour plus de simplicité, ils seront noté MCSM dans la suite de ce document.

7.1.2 Analyse du jeu de handball par modèles segmentaux à partir des interactions entre trajectoires de joueurs

Les modélisations proposées dans la section précédente pour l'interprétation de vidéos de squash sont étendues à l'analyse de trajectoires issues de vidéos de handball. Ces dernières sont reconstruites dans le plan du terrain de jeu, et incluent les trajectoires de l'ensemble des joueurs d'une même équipe.

7.1.2.1 Caractérisation de l'interaction entre les joueurs de handball

Nous partons des trajectoires reconstruites dans le plan du terrain de handball, et ce à l'aide de deux caméras panoramiques filmant chacune une des deux moitiés du terrain. Les méthodes de suivi qui ont permis la reconstruction, dans le plan du terrain, des trajectoires sont d'après [CVBASEDOC] des techniques "classiques" de vision par ordinateur. La figure 8.6 contient un "bout" de trajectoire, pour les sept joueurs d'une même équipe, pour une durée d'une vingtaine de secondes de jeu.

La représentation des différentes interactions entre les joueurs de handball ainsi que leurs dynamiques résulte de trois considérations principales, qui sont les suivantes :

- concentrer l'information d'interaction sur un nombre limité de descripteurs, une représentation exhaustive des distances entre joueurs de handball posant notamment le problème du coût de calcul,
- exploiter le rôle très spécifique du gardien de handball,
- prendre en compte la dynamique globale des joueurs de champ.

Ainsi, nous avons retenu cinq variables devant permettre de prendre en compte les trois points énoncés ci-dessus, à savoir, à chaque instant t :

- la distance moyenne entre le gardien et les six joueurs de champ $d_{GC,t}$,

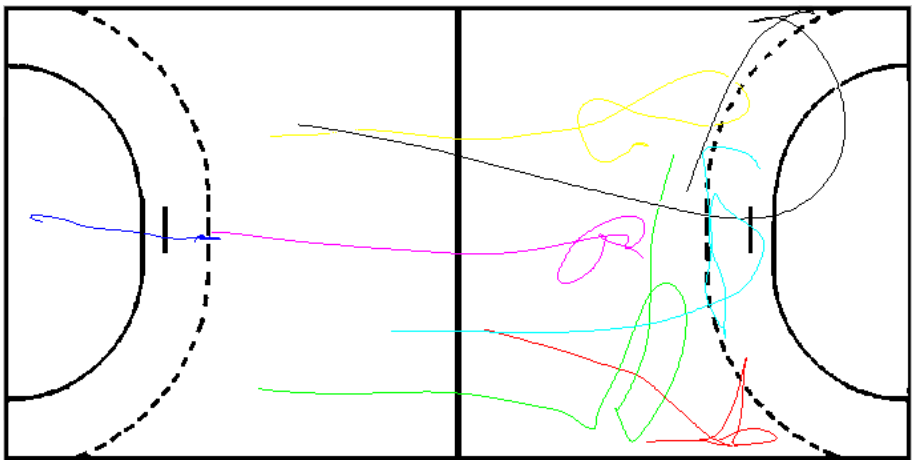

FIG. 7.4 – Représentation d'une portion de trajectoire (500 points), pour les sept joueurs de handball d'une même équipe, gardien compris, lors d'un retour en défense.

- la distance moyenne entre les six joueurs de champ $d_{C,t}$,
- le minimum, la moyenne et le maximum des distances entre positions successives (*i.e.*, distance parcourue entre $t-1$ et t) des six joueurs de champ, notée $d_{intramin,t}$, $d_{intramean,t}$ et $d_{intramax,t}$.

Les trois dernières valeurs $d_{intramin,t}$, $d_{intramean,t}$ et $d_{intramax,t}$ permettent d'avoir une connaissance globale, à tout instant t, sur la dynamique des joueurs de champ.Ainsi, la représentation, à un instant t, est composée au total de :

$$d_{GC,t}, d_{C,t}, d_{intramin,t}, d_{intramean,t}, \text{ et } d_{intramax,t}.$$

Les cinq vecteurs représentant les interactions entre joueurs de handball ainsi que leurs dynamiques propres, pour chaque phase de jeu, seront donc composés des valeurs successives (une pour chaque image) de $d_{GC,t}$, $d_{C,t}$, $d_{intramin,t}$, $d_{intramean,t}$, et $d_{intramax,t}$. Par exemple, pour les valeurs de $d_{GC,t}$, nous avons

$$D_{GC} = [d_{GC,1}, ..., d_{GC,n-1}, d_{GC,n}].$$

Les trajectoires étant reconstruites dans le plan du terrain de handball, il n'est pas besoin d'avoir une représentation invariante, au contraire de la section précédente dans laquelle les trajectoires étaient directement extraites de la vidéo de squash, et non reconstruites dans le plan de jeu.

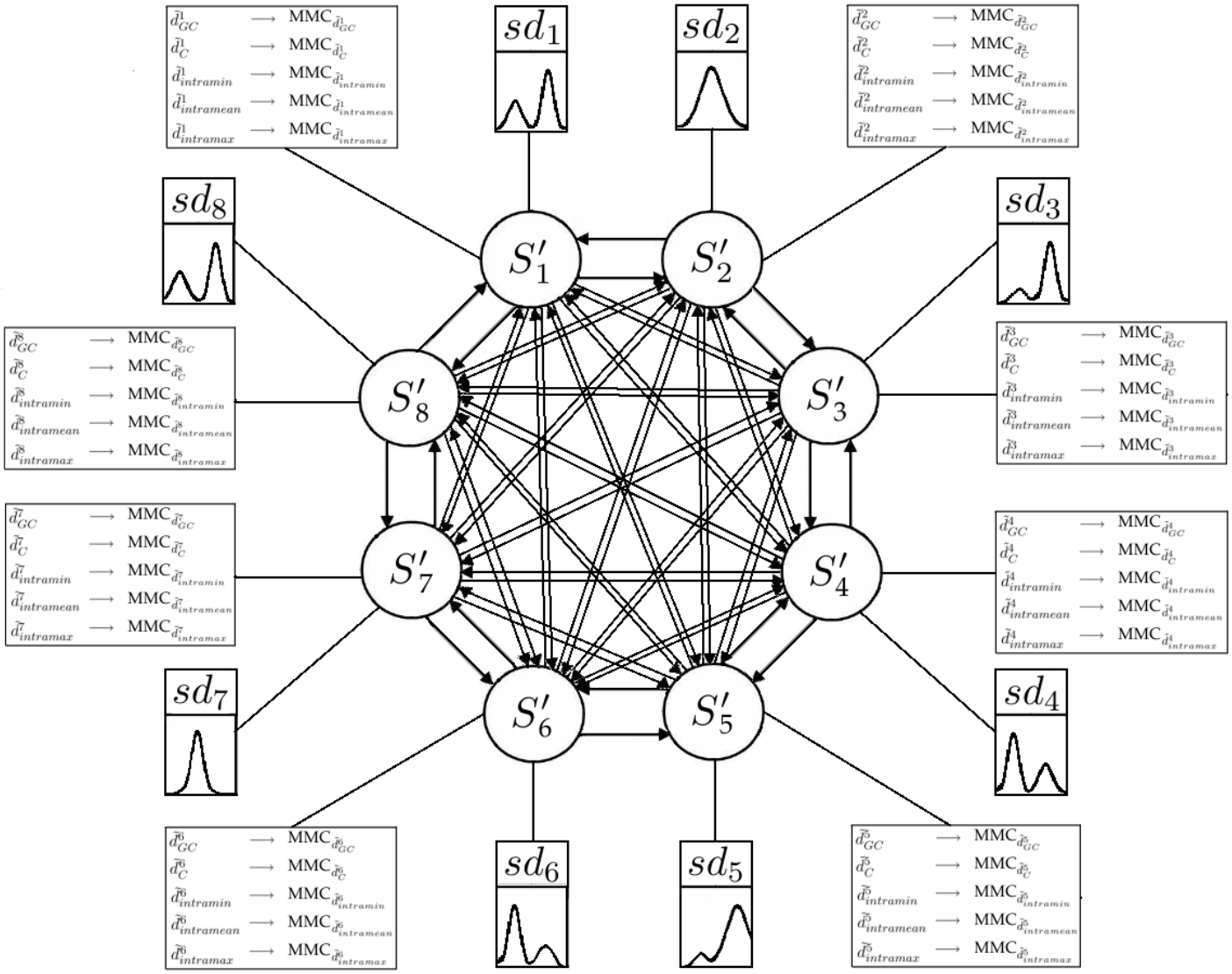

FIG. 7.5 – Modélisation par MCSM du jeu de handball, avec 8 états S'_i correspondant chacun à une phase précise de l'activité des joueurs. Chacune de ces phases d'activité est représentée par un MMC Pa à cinq composantes, une pour chaque descripteur, ainsi que par un MMG modélisant les temps de séjour de séjour (sd_i est le temps de séjour associé à la phase d'activité S'_i).

Pour réduire le temps de calcul, un regroupement des données est ajouté. Pour les cinq vecteurs de représentations (*i.e.*, D_{GC}, D_C, $D_{intramin}$, $D_{intramean}$ et $D_{intramax}$), des groupes de k_{group} données consécutives sont formés. Les valeurs moyennes de chacun de ces groupes, pour chacun des cinq vecteurs de représentation, sont calculées et nous formons de nouveaux vecteurs de représentation $\tilde{D}_{GC}$, $\tilde{D}_C$, $\tilde{D}_{intramin}$, $\tilde{D}_{intramean}$ et $\tilde{D}_{intramax}$ (de taille k_{group} fois inférieure).

7.1.2.2 Modélisation des phases du jeu de handball par modèles cachés semi-markoviens

Nous allons exploiter les MCSM développés en section 7.1.1.3 pour modéliser les activités observées dans un match de handball. La modélisation mise en place est largement similaire à celle proposée pour la reconnaissance de phases de jeu dans des vidéos de squash, en utilisant les cinq descripteurs développés pour les trajectoires de handball ainsi que huit phases d'activités décrites dans la suite. Les représentations choisies (*i.e.*, $d_{GC,t}$, $d_{C,t}$, $d_{intramin,t}$, $d_{intramean,t}$, et $d_{intramax,t}$). Nous avons mis en place un schéma à deux niveaux. Les descripteurs des trajectoires et des interactions entre trajectoires sont modélisées à l'aide des MMC/MMG Pa. Ces modélisations par MMC/MMG Pa permettent de caractériser les états du modèles cachés semi-markoviens (MCSM).

Les états du MCSM, les S'_i, forment la couche de niveau supérieur de cette modélisation. La couche de niveau inférieur est formée par les MMC/MMG Pa permettant de prendre en compte les cinq variables de description des trajectoires. La figure 7.5 contient une illustration de la modélisation hiérarchique mise en place.

⋆ *Modélisation des activités par MCSM*

Les états S'_i du MCSM décrivent les phases d'activité. Les huit phases d'activité pour le jeu de handball sont les suivantes :
- "montée lente en attaque",
- "attaque placée",
- "arrêt de jeu offensif" (remise en jeu au neuf mètres adverse),
- "arrêt de jeu défensif" (remise en jeu au neuf mètres),
- "retour lent en défense",
- "défense placée",
- "retour rapide en défense" (face à une contre-attaque),
- "attaque rapide" (contre-attaque).

Dans la modélisation MCSM mise en place, des MMG sont utilisés pour modéliser les distributions des temps de séjour sd_i associés à ces phases d'activité.

Des procédures de type "forward-backward" (voir section 2.3) sont utilisées pour déterminer les paramètres des MMG. Les initialisations de ces procédures sont effec-

tuées en exploitant des algorithmes de type "k-means" (voir section 2.4). L'ensemble des paramètres associé à la modélisation des temps de séjour sd_i par MMG est noté ψ.

⋆ *Modélisation des représentations à l'aide de MMC Pa*

Les descripteurs de trajectoires et de leurs interactions $d_{GC,t}$, $d_{C,t}$, $d_{intramin,t}$, $d_{intramean,t}$, et $d_{intramax,t}$ sont utilisés pour caractériser chacune des huit phases d'activité S_i. Ils sont notés dans la figure 7.5, pour une phase d'activité S'_i, d^i_{GC}, d^i_C, $d^i_{intramin}$, $d^i_{intramean}$, et $d^i_{intramax}$. Pour chaque état de niveau supérieur, la modélisation de ces descripteurs est effectuée par MMC/MMG Pa (voir section).

Le choix du nombre d'états des composantes MMC/MMG des MMC/MMC Pa est effectué de manière analogue à celle décrite en section 7.1.1.3.

⋆ *Entraînement et algorithme de Viterbi pour la reconnaissance de phases de jeu dans des vidéos de handball*

Soit une séquence d'états S' contenant R segments et, donc, R phases d'activités successives, chacune associée à un unique état S'_i. Soit également q_r l'index temporel du temps de fin du r^{me} segment, les observations du r^{me} segments sont $y_{(q_{r-1}+1,q_r]} = y_{q_{r-1}+1}, \ldots, y_{q_r}$ telles que $S'_{q_{r-1}+1} = \ldots = S'_{q_r}$. S' correspond à la séquence d'états S'_i des MCSM. A' est la matrice de probabilité de transitions entre états de niveau supérieur des MCSM aux temps $\{q_i\}$, calculée à l'aide des transitions entre phases d'activité observées dans la vidéo d'apprentissage.

Les trajectoires extraites des vidéos d'entraînement sont exploitées afin de déterminer les paramètres des MMC/MMG Pa. L'ensemble des paramètres, pour toutes les composantes des MM/MMG Pa est noté ϕ. Il est composé des paramètres des MMC/MMG (B, A et π) pour chacune des cinq composantes des MMC/MMG Pa, et ce pour chaque état de niveau supérieur S'_i.

L'ensemble des paramètres est ainsi défini par $\theta = \{A', \phi, \psi\}$ et est estimé par apprentissage supervisé. Ces paramètres permettent d'effectuer la reconnaissance des phases d'activité temporelles. L'interprétation des vidéos de handball est alors réalisée à l'aide d'un algorithme de décodage de Viterbi pour MCSM (voir sections B.1 et 1.3.6).

L'algorithme de Viterbi permet de trouver la séquence d'états de niveau supérieur $\hat{S}$ des MCSM maximisant la log-vraisemblance,*i.e.*, telle que $\hat{S}' = \arg\max_{S'} \log P(y, S'|\theta)$. La vraisemblance $P(y, S'|\theta)$ est définie, pour une séquence d'observations y, et la séquence d'états de MCSM correspondante S', par :

$$
\begin{aligned}
P(y, S'|\theta) &= \prod_{r=1}^{R} P(S'_r|S'_{r-1}) \\
&\times \prod_{r=1}^{R} P(sd_i = q_r - q_{r-1}|\psi; S'_{q_r}) \\
&\times \prod_{r=1}^{R} P(y_{(q_{r-1},q_r]}|\phi; S'_{q_r}).
\end{aligned}
$$

⋆ Intégration de données sonores : prise en compte automatique des coups de sifflet

Dans le domaine du multimédia, de nombreuses méthodes ont été proposées pour le traitement simultanée de données visuelles et sonores pour la reconnaissance de contenu. Une étude poussée des descripteurs audiovisuels de bas-niveau, de l'analyse de la structure pour des contenus audiovisuels, et de solutions d'interopérabilité pour l'indexation audio et vidéo sont disponibles dans [Joly 06]. Nous citons ici quelques exemples des traitements audiovisuels pour la reconnaissance de scènes. Ainsi, Leonardi et al. ont proposé des méthodes pour l'analyse de documents multimédia. Par exemple, dans [Leonardi 02], ils ont considéré des scènes extraites de vidéos et cherchent à identifier leur type parmi les suivants : dialogue, histoire, action, générique. Chaque type de scène est modélisé par un MMC, les vecteurs d'observations étant constitués de descripteurs audios et de descripteurs de mouvements extraits du flux MPEG. Au contraire, dans des travaux proposés par Wang et al. [Wang 00], un MMC est appris pour chaque type de descripteurs. La classification des scènes est ensuite réalisée par fusion de ces modèles en calculant le produit des vraisemblances de chaque type d'observation relativement à chacun des MMC correspondant.

Des travaux ont également été menés, notamment par des laboratoires suisses, pour la compréhension de scènes de réunion à l'aide de document audiovisuel. Dans [Hung 07], les auteurs ont proposé une méthode pour la détection de personne dominante dans des réunions. Ils se sont appuyés sur une représentation des documents formée de quatre caractérisation, deux d'entre elles exploitant des informations sonores, les deux autres étant extraites d'informations visuelles. Zhang et al. ont également proposé des modélisations, tant pour les actions individuelles que de groupes, à l'aide de documents audiovisuels. Afin de prendre en compte les interactions observées entre individus, ils ont mis en place une modélisation par MMC à deux niveaux, le premier niveau modélisant les comportements individuels et le second intégrant des informations liées au groupe pour la compréhension des actions [Zhang 04].

Au vu des premiers résultats obtenus et afin d'obtenir des résultats précis de reconnaissance de phases de jeu, nous avons décidé d'exploiter des données sonores. En effet, le flux sonore correspondant à la vidéo de handball est disponible, permettant de prendre en compte une information importante dans la reconnaissance des différentes phases d'activité : les coups de sifflet de l'arbitre.

L'analyse du flux sonore est faite par requête, un segment sonore d'apprentissage (*i.e.*, le segment sonore correspondant à un coup de sifflet) est utilisé. Les instances de coup de sifflet sont retrouvés dans le flux sonore à l'aide de deux logiciels développés par l'équipe-projet (de l'INRIA Rennes - Bretagne Atlantique) Metiss : SPro et Audioseg, disponibles en ligne (aux adresses [SPro] et [Audioseg]). Spro produit la description, d'une part de la requête (le flux sonore associé aux coups de sifflet), et d'autre part de l'ensemble du flux sonore associé à la vidéo observée. La description des signaux sonores s'appuie sur les coefficients cepstraux de fréquence mel [Mermelstein 76], un ensemble de coefficients étant défini pour chacun des intervalles décrits par une fenêtre glissante. Audioseg est ensuite utilisé pour la reconnaissance de la requête à l'intérieur de l'ensemble du flux sonore [Muscariello 09]. Elle repose sur une procédure de type DTW (voir section 3.1.1).

Cette information est utilisée d'une façon très simple par la méthode MCSM, *i.e.*, par le biais d'une hypothèse de changement d'état de niveau supérieur S'_i dès qu'un coup de sifflet est détecté. Cette information est prise en compte dans le décodage de Viterbi de façon simple. Nous postulons que, dès qu'un coup de sifflet est détecté, la phase de jeu est arrêtée. Cela permet d'obtenir une partition de l'action observée en segments successifs Seg_k, un segment étant le temps écoulé entre deux coups de sifflet. Ensuite, chacun des segments est décodé (*i.e.*, interprété) séparément par l'algorithme de Viterbi, avec pour seule contrainte que le premier état de niveau supérieur S'_i d'un segment Seg_{l+1} soit différent du dernier état de niveau supérieur S'_j trouvé par l'algorithme de Viterbi pour le segment Seg_l.

7.1.3 Une méthode pour comparaison : les modèles de Markov cachés hiérarchiques

Les modèles de Markov cachés hiérarchiques parallèles (voir section 1.3.5) sont inspirés de la méthode MCSM développés en sections 7.1.1.3 et 7.1.2, utilisés pour l'interprétation de vidéos de sport. Pour des raisons de clarté, les modèles de Markov cachés hiérarchiques parallèles définis dans cette section seront, dans la suite du document, notés MMCH.

La différence, en comparaison des MCSM, est que les temps de séjour ne sont pas modélisés dans les MMCH (voir figure 7.6). Ainsi, contrairement aux modélisations

semi-markoviennes, le temps de séjour est guidé par une loi géométrique telle que :

$$p(d_i) = a_{ii}^{d-1}(1 - a_{ii}).$$

Nous allons appliquer cette méthode à des fins de comparaison pour les tâches reconnaissance de phases de jeu dans des vidéos de squash et de handball.

L'apprentissage du modèle MMCH est similaire à celui des MCSM, sauf pour la matrice A. En effet, pour le modèle MCSM, la matrice A est calculée à partir des transitions entre phases d'activité observées dans la vidéo d'apprentissage. Pour les modèles MMCH, la matrice A sera calculée en considérant chacune des images observées et son appartenance à une phase d'activité donnée. Ainsi, les phases d'activités étant composées d'un nombre d'images important, les valeurs de la diagonale de A, les a_{ii}, auront des valeurs proches de 1 alors que les probabilités de changement de phases à un instant donné, les $a_{ij}, i \neq j$, auront des valeurs proches de 0. L'interprétation des vidéos de handball est réalisée à l'aide d'un algorithme de Viterbi (voir section 1.3.5)

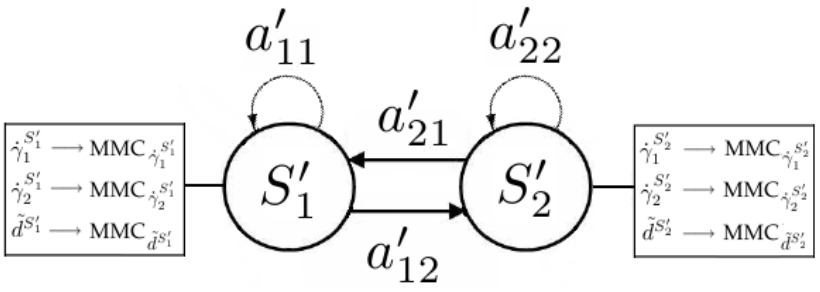

FIG. 7.6 – Modélisation par MMCH du jeu de squash, avec 2 états de niveau supérieur (S'_1 et S'_2) correspondant chacun à une phase précise de l'activité des joueurs ("jeu" et "non-jeu"). Chacune de ces phases d'activité est représentée par un MMC Pa à trois composantes, une pour chaque descripteur.

7.2 Conclusion

Nous avons introduit différentes modélisations des interactions entre trajectoires et les avons mises à profit pour la caractérisation de phases de jeu dans des vidéos de sport. Les MMC/MMG Pa sont ensuite utilisés au sein d'une modélisation semi-markovienne originale pour l'interprétation de phases d'activité. En effet, les MMC/-MMG Pa permettent de modéliser les différentes composantes des représentations proposées. Les méthodes semi-markovienne sont utilisées afin de prendre en compte les temps de séjour dans les phases de jeu. La pertinence de cette dernière information prise en compte pourra être mise en valeur par la comparaison avec les modélisations markoviennes hiérarchiques. Cette approche a été appliquée à la reconnaissance de phases de jeu dans des vidéos de squash et de handball.

Chapitre 8

Applications

"[...] j'ai dit que la complexité, c'est l'union de la simplicité et de la complexité ; c'est l'union des processus de simplification qui sont sélection, hiérarchisation, séparation, réduction, avec les autres contre-processus qui sont la communication, qui sont l'articulation de ce qui est dissocié et distingué ; et c'est d'échapper à l'alternative entre la pensée réductrice qui ne voit que les éléments et la pensée globaliste qui ne voit que le tout.
Comme disait Pascal : « Je tiens pour impossible de connaître les parties en tant que parties sans connaître le tout, mais je tiens pour non moins impossible la possibilité de connaître le tout sans connaître singulièrement les parties »."

Edgar Morin - ***Introduction à la pensée complexe***

Nous allons étudier une forme d'analyse sémantique de vidéos de sport. Les données utilisées sont issues d'une base de données [CVBASEDATA] contenant des trajectoires extraites de vidéos de squash, ainsi que des trajectoires extraites de vidéos de handball et reconstruites dans le plan du terrain. Nous allons décrire les résultats expérimentaux obtenus tour à tour sur ces deux types de sport, le squash et le handball. L'objectif est de reconnaître différentes phases d'activité dans ces deux sports.

Rappelons dans ce début de chapitre que les notations **reconnaissance de phases de jeu** ou **interprétation** de vidéos de sport correspondent aux tâches, effectuées de façon simultanée, de segmentation temporelle et de reconnaissance des phases de jeu dans des vidéos de sport.

8.1 Reconnaissance de phases de jeu dans des vidéos de squash

Pour tester les méthodes d'interprétation de vidéos de sport, nous nous intéressons tout d'abord à des vidéos de squash. Ces vidéos sont filmées par une caméra placée au dessus de l'action, au milieu du terrain. Pour cela, nous exploitons des trajectoires de joueurs fournies dans [CVBASEDATA]. Elles ont été extraites d'une vidéos de plus de dix minutes de squash composée de 15508 images (voir en figure 8.1 une image tirée de la vidéo de squash). Les trajectoires sont données par les coordonnées spatiales respectives des deux joueurs observés dans les images. Nous disposons aussi des phases d'activité qui relèvent en fait de deux classes : phases de "jeu" et de "non-jeu" [CVBASEDOC]. Elles sont utilisées comme vérité terrain pour l'évaluation de notre méthode.

FIG. 8.1 – Deux images tirées de la vidéo de squash (la vidéo entière traitée comprend 15508 images). À gauche une image appartenant à la phase "jeu", à droite une image appartenant à la phase "non-jeu".

8.2 Expérimentations réalisées

La première moitié de la vidéo de squash (correspondant à 7422 images) a été utilisée pour apprentissage du MSMC et également du MMCH (définis en section 7.1.2 et 7.1.3). Les deux états de niveau supérieur S'_1 et S'_2 correspondent aux deux phases d'activité "jeu" et "non-jeu". La seconde moitié de la vidéo (correspondant à 8086 images) et les trajectoires correspondantes ont été utilisées comme données de test pour évaluer notre méthode. La figure 8.2 contient les trajectoires des deux joueurs de squash, respectivement dans les images utilisées pour l'apprentissage et dans la partie utilisée pour les tests. Tous les résultats ont été obtenus avec une valeur de k_{group} égale à 8.

La phase “jeu” est définie par les périodes entre le début d’un point (le service) et la fin d’un point (la balle rebondit deux fois par terre et le point est fini). La phase “non-jeu” correspond aux périodes entre la fin d’un point et le début du point suivant. Ainsi, la phase non-jeu contient une activité importante. En effet, entre deux points, l’un des deux joueurs doit aller chercher la balle, puis les deux joueurs peuvent échanger leurs places selon que le serveur a gagné ou perdu le point. De plus, il est important de souligner que dans la vidéo traitée, les joueurs sont des professionnels. Ces joueurs se placent donc de façon optimale sur le terrain, au contraire de joueurs débutants ou amateurs qui peuvent avoir tendance à se placer de manière désordonnée sur le terrain. On observe ainsi que les joueurs professionnels ont assez peu de mouvement pendant les phases de “jeu”.

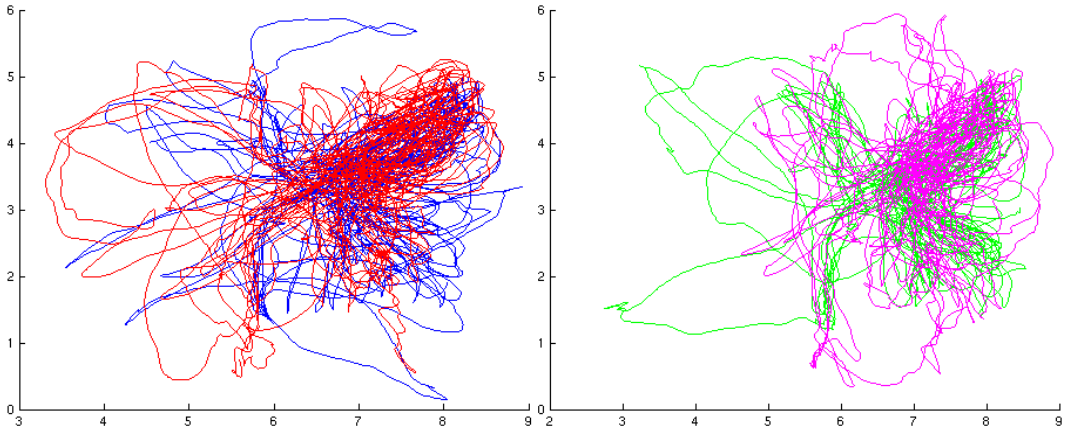

FIG. 8.2 – A gauche : les trajectoires extraites des images test (en bleu et en rouge) sur la seconde partie de la vidéo de squash (au total 8086 images). A droite : les trajectoires des deux joueurs de squash utilisées pour l’apprentissage (en vert et en magenta) sur la première partie de la vidéo de squash (au total 7422 images).

Les expériences réalisées comportent donc une difficulté importante. En effet, avec les seuls mouvements des joueurs (sans prise en compte des mouvements de la balle ou des mouvements de raquette), il est compliqué, visuellement, de savoir si les joueurs sont en phase “jeu” ou “non-jeu”. La quantité de mouvements des joueurs en phase “jeu” ne s’avère pas si différente de celle de la phase “non-jeu”. Les joueurs ne “bougent” donc pas toujours plus en phases de jeu qu’en phases de non-jeu, contrairement à ce que l’on pourrait croire *a priori*. Cela sera illustré dans les résultats décrits ultèrieurement.

Les résultats de reconnaissance de phases de jeu présentés dans la suite correspondent aux ratios entre le nombre d’images classées correctement (parmi les phases

"jeu" et "non-jeu") et le nombre d'images traitées. Pour cela, la vérité-terrain connue pour l'ensemble des trajectoires test est utilisée, une image test étant correctement classée si l'interprétation obtenue attribue à cette image la même phase de jeu que celle attribuée par la vérité-terrain.

8.3 Résultats obtenus

La première moitié de la vidéo de squash (correspondant à 7422 images) a donc été utilisée pour apprentissage du MSMC illustré dans la figure 7.3. Elle a permis de déterminer les paramètres $\theta = \{A', \phi, \psi\}$ définis dans la section 7.1.2. La figure 8.3 présente la modélisation par MMG de la distribution des temps de séjour observés pour l'état correspondant à la phase "jeu".

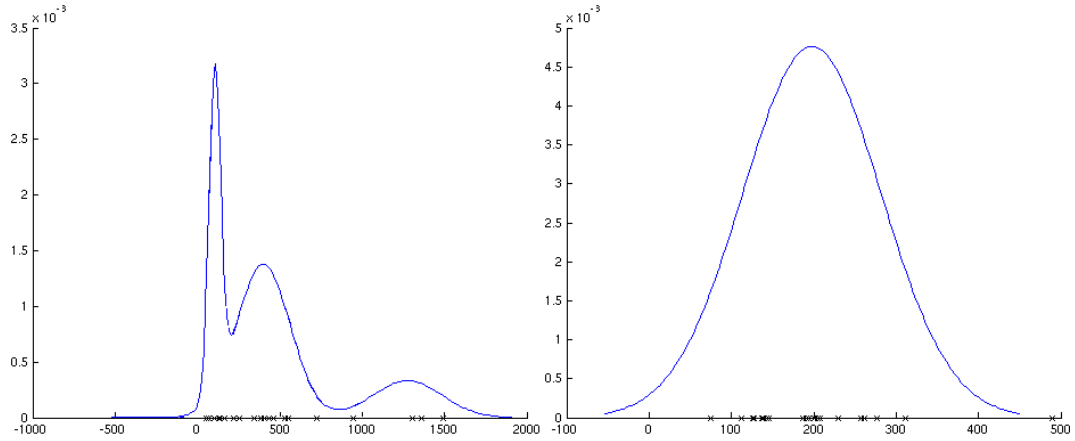

FIG. 8.3 – À gauche : Modélisation par un MMG du temps de séjour pour l'état de niveau supérieur du MCSM correspondant à la phase d'activité "jeu". L'axe des abscisses correspond au temps de séjour (les croix indiquant les temps de séjour observés). À droite : Modélisation par un MMG du temps de séjour pour l'état de niveau supérieur du MCSM correspondant à la phase d'activité "non-jeu".

Un taux de 89.2% d'images correctement classées a été obtenu sur les 8086 images tests avec la méthode MCSM. La figure 8.4 présentent notamment les résultats de reconnaissance obtenus par la méthode MCSM, phases par phases, pour les images test de la seconde partie de la vidéo observés.

la méthode MMCH conduit à des résultats de 88% de reconnaissance des phases de jeu. La figure 8.4 présente également l'interprétation obtenue par la méthode MMCH pour les images de la seconde partie de la vidéo de squash.

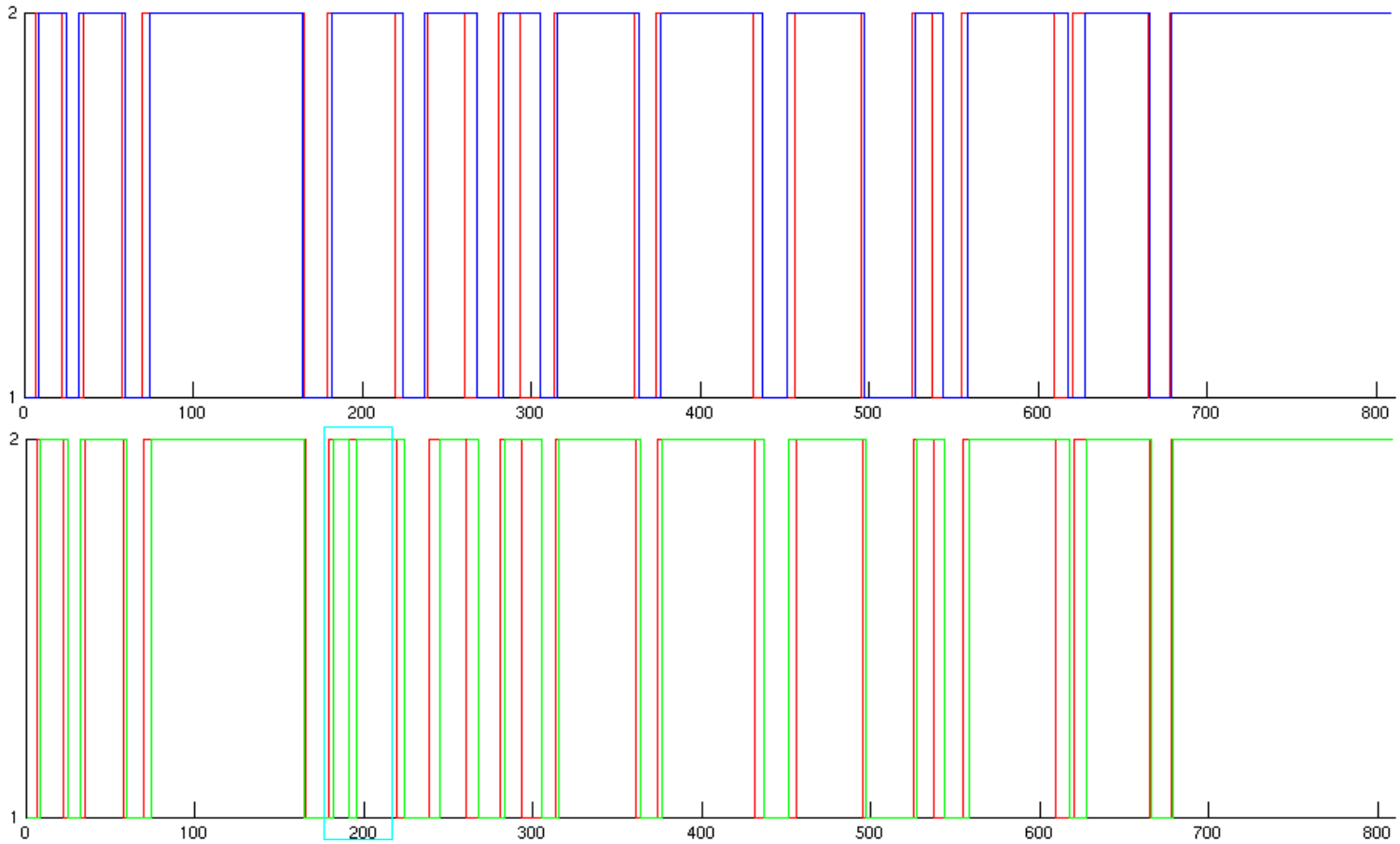

FIG. 8.4 – En haut : résultats d'interprétation obtenus par notre méthode MCSM sur la seconde partie de la vidéo de squash (au total 8086 images). Les valeurs "1" et "2", en ordonnée, correspondent respectivement aux phases "non-jeu" et "jeu". La vérité-terrain est tracée en rouge, l'interprétation obtenue lui étant superposée en bleu. Ainsi, lorsque des segments rouges apparaissent, cela correspond en fait à des imprécisions temporelles de reconnaissance. En bas : résultats d'interprétation de vidéos de squash obtenus par la méthode MMCH sur la seconde partie de la vidéo de squash. Les résultats obtenus sont tracés en vert, superposés à la vérité-terrain en rouge. La partie entourée en bleu ciel contient la principale différence d'interprétation constatée entre la méthode MMCH et la méthode MCSM.

Nous avons également mené des expérimentations en ne tenant pas compte de la distance entre les joueurs décrite par le vecteur D. Ainsi, nous avons considéré une version de la méthode MCSM appliquée aux seuls vecteurs V_1 et V_2. Cette dernière n'inclut donc que la prise en compte du mouvement des deux joueurs, les interactions entre les deux joueurs étant prise en compte par D. Les résultats d'interprétation corrects obtenus, dans les mêmes conditions d'apprentissage et de test, passe de 88% à moins de 70%.

Enfin nous avons testé la méthode MCSM en utilisant, pour caractériser les phases de jeu, seulement le vecteur D. Les vecteurs V_1 et V_2 sont alors ignorés et seule l'interaction entre les joueurs est prise en compte. Le résultat d'interprétation correcte de vidéos de squash obtenus, sur les 8086 images tests, est de 86.8%. Il est donc légèrement inférieur à celui de 88% obtenu en exploitant D, V_1 et V_2.

8.4 Description et commentaires des résultats

Les résultats obtenus avec la méthode MCSM sont satisfaisants puisque, au delà du pourcentage de bonne reconnaissance de 89.2%, on note que toutes les phases de jeu ont été parfaitement détectées. En effet, les transitions entre les phases "jeu" et "non-jeu" ont été détectées, sans erreurs. Les 10.8% d'imprécisions de reconnaissance correspondent à des décalages de la reconnaissance réalisée aux débuts et aux fins des phases d'activité. Ainsi, la méthode a exactement détecté les 13 points (*i.e.*, les 13 phases de "jeu") observés.

La comparaison entre la méthode MCSM et la méthode MMCH a également permis de montrer la pertinence de la modélisation MCSM. La méthode MMCH réalise une reconnaissance de phases moins précise de 1.2%. De plus, la méthode MMCH réalise une interprétation en phases de jeu imparfaite. En effet, là ou la méthode MCSM détecte exactement les phases de jeu (c'est-à-dire les 13 points observés), la méthode MMCH détecte une phase de "jeu" non observée. L'interprétation obtenue avec la méthode MMCH présente donc 14 points joués au lieu de 13. Ceci est illusté dans la partie inférieure de la figure 8.4, et notamment dans le cadre bleu-ciel, on constate qu'une phase de "jeu" non observée dans la vérité-terrain est détectée. Cela illustre l'intérêt de prendre en compte les temps de séjour dans les états de niveau supèrieur (ici, S'_1 et S'_2).

Enfin, les résultats obtenus en considérant la méthode MCSM n'incluant que la prise en compte du mouvement des deux joueurs, au travers des vecteurs V_1 et V_2, montrent l'importance d'appréhender explicitement l'interaction entre les deux joueurs. Les trajectoires individuelles des joueurs ne permettent pas de retrouver efficacement les phases de "jeu" et de "non-jeu". Cela confirme les observations faites dans l'introduction en section 8.2.

De plus, la comparaison avec la méthode MCSM ne prenant pas en compte les valeurs des descripteurs $\dot{\gamma}$ semble montrer que l'exploitation de la dynamique et de la forme des trajectoires de chacun des joueurs (à l'aide V_1 et V_2) aide à obtenir une reconnaissance plus précise. Néanmoins, comme il est précisé en section 8.6, utiliser seulement le vecteur D permet d'avoir des temps de calcul largement inférieurs. En effet, les temps de calculs des approximations de trajectoires nécessaires afin d'obtenir les descripteurs $\dot{\gamma}$ sont importants.

8.4.1 Conclusion

Nous avons mis en évidence l'intérêt de formaliser les interactions entre joueurs pour le traitement de trajectoires extraites de vidéos de squash. En effet, la méthode MCSM proposée a permis d'obtenir des résultats d'interprétation de vidéos de squash satisfaisants, permettant de reconnaître précisément les phases de jeu et celles de non-jeu. La représentation invariante des trajectoires, à l'aide des variables $\dot{\gamma}$ et des distances entre les deux joueurs, doit permettre de traiter toute trajectoire extraite de vidéos de squash, sans avoir le besoin de ré-entraîner le système.

8.5 Reconnaissance de phases de jeu dans des vidéos de handball

Nous avons eu recours aux trajectoires de l'ensemble des joueurs de handball d'une même équipe disponibles dans la base de données [CVBASEDATA]. Elles ont été extraites de deux vidéos filmées par deux caméras. Les deux caméras sont placées au dessus de l'action, une caméra étant disposée au milieu de chaque moitié du terrain de handball. Les trajectoires ont en fait été reconstruites dans le plan du terrain de handball. La figure 8.5 présente trois images prises au même moment du match de handball, deux images sont extraites des deux caméras filmant l'action du dessus, la troisième est issue d'une caméra sur le bord du terrain. La figure 8.6 présente l'ensemble des trajectoires considérées, pour les sept joueurs d'une même équipe, et ce pour l'ensemble des dix minutes de jeu traitées (plus précisément 9 minutes et 47 secondes, les vidéos correspondant à une séquence de 14664 images avec 25 images par seconde).

La base de données fournit également une segmentation (à l'aide de différentes phases de jeu) que nous avons utilisée afin de construire la vérité-terrain à partir des huit phases de jeu définies en section 7.1.2. Les huit phases de jeu définies dans le chapitre précédent sont quelque peu différentes de celles considérées dans la base de données. Comme nous l'avons également souligné dans la section 7.1.2.2, les coups de sifflet étant des indicateurs fiables de changement de phases de jeu, ils ont été exploités pour la construction de la vérité-terrain. En effet, l'hypothèse de changement de

FIG. 8.5 – Trois images correspondant à un même instant. En haut, les images ont été prises par deux caméras, une au dessus de chaque moitié de terrain. En bas, une image prise par une caméra sur le bord du terrain.

phases à chaque coup de sifflet a été vérifiée. La segmentation disponible dans la base de données confirme bien l'existence de changements de phases de jeu à chaque coup de sifflet, à quelques légers décalages près.

Les phases de jeu considérées sont définies avec leur numérotation comme suit :

- "montée lente en attaque" : phase d'action 1,
- "attaque placée" : phase d'action 2,
- "arrêt de jeu offensif" (remise en jeu au neuf mètres adverse) : phase d'action 3,
- "attaque rapide" (contre-attaque) : phase d'action 4,
- "retour rapide en défense" (face à une contre-attaque) : phase d'action 5,
- "retour lent en défense" : phase d'action 6,
- "défense placée" : phase d'action 7,
- "arrêt de jeu défensif" (remise en jeu au neuf mètres) : phase d'action 8.

Les numérotations présentées sont utilisées, dans la suite, dans les figures illustrant les résultats d'interprétation obtenus.

Nous décrivons maintenant les tests comparant les méthodes MCSM et MMCH développées pour la reconnaissance des phases de jeu dans des vidéos de handball à partir des trajectoires des joueurs d'une même équipe (voir section 7.1.2).

Il nous faut tout d'abord souligner que les trajectoires disponibles (les trajectoires des sept joueurs d'une même équipe pendant dix minutes de jeu) constituent un ensemble de données de relativement faible taille dans la perspective de l'apprentissage des modèle semi-markoviens hiérarchiques parallèles (MCSM) et des modèles de Markov cachés hiérarchiques parallèles (MMCH) définis en section 7.1.2. En effet, ces modèles, complexes dans leurs structures, nécessiteraient des trajectoires de joueurs de handball sur une plus longue durée. Des trajectoires correspondant à au moins quelques heures de jeu seraient nécessaires afin d'appréhender la variabilité et la diversité des comportements de jeu.

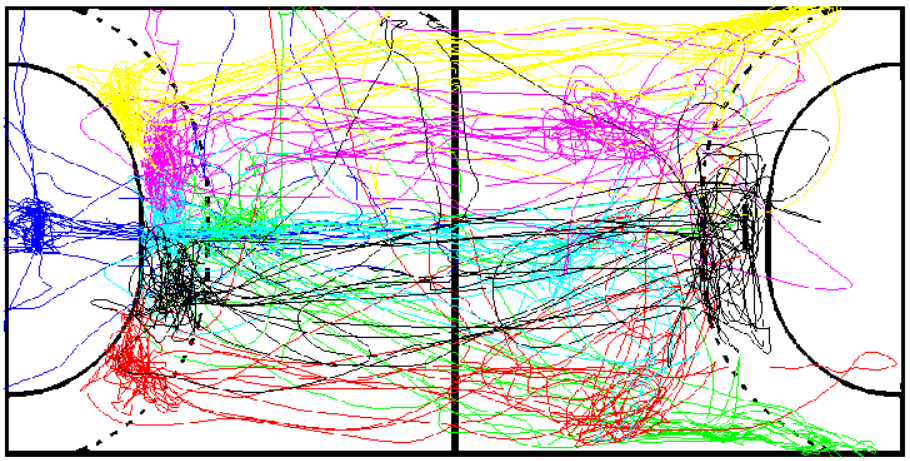

FIG. 8.6 – Tracé des trajectoires, reconstruites dans le plan du terrain de handball, des sept joueurs d'une même équipe de handball (chaque trajectoire, associée à un joueur, est d'une couleur particulière), et ce pour les dix minutes de jeu.

Ainsi, pour l'apprentissage de la matrice de transition entre états de niveau supérieur A' ainsi que pour celui des MMG modélisant les temps de séjour dans les états de niveau supérieur S'_i, des observations extraites de différents matches de handball ont été utilisées. En effet, ces paramètres ne requièrent pas l'obtention préalable des trajectoires afin d'être appris. Ainsi, en observant et en segmentant, à l'aide des huit phases de jeu, un programme TV de handball (correspondant à une heure de handball), nous avons pu construire des données supplémentaires en termes de transitions

entre phases de jeu et de temps de séjour dans les phases de jeu. Ces données ont notamment été extraites de la finale des jeux olympiques de Pékin, en août 2008. Les vidéos correspondantes peuvent notamment être trouvées en ligne [Dailymotion].

Les résultats d'interprétation correcte de vidéos de handball présentés dans la suite correspondent aux ratios entre le nombre d'images classées correctement (parmi les huit phases de jeu) et le nombre d'images traitées. Pour cela, la vérité-terrain disponible sur l'ensemble des trajectoires test est exploitée, une image test étant correctement classée si l'interprétation obtenue attribue à cette image la même phase de jeu que celle attribuée par la vérité-terrain. Tous les résultats présentés dans cette section ont été obtenus avec une valeur de k_{group} égale à 8 (voir section 7.1.2). Enfin, nous avons initialisé le système en lui indiquant que la première phase rencontrée au moment du coup d'envoi est soit la phase "montée lente en attaque", soit la phase de jeu "défense placée". En effet, au début d'un match de handball, une équipe ne peut se trouver que dans l'une de ces deux phases de jeu.

8.5.1 Expérimentations réalisées et résultats obtenus

Nous avons dans un premier temps considéré l'ensemble des trajectoires de handball disponibles, c'est-à-dire les trajectoires des sept joueurs d'une même équipe sur les dix minutes de jeu, pour la phase d'apprentissage du modèle. Puis nous avons cherché à reconnaître les phases de jeu sur ce même ensemble de données.

Ces tests ont été menés pour évaluer la pertinence du modèle proposé dans les conditions les plus favorables possibles. Le résultat de reconnaissance de phases de jeu obtenu, présenté à la figure 8.7, à partir des seules trajectoires (sans la détection de coups de sifflet conduisent à un taux de 89% d'interprétation correcte).

Nous avons également effectué des tests en utilisant la première partie des trajectoires disponibles (correspondant à 6370 images) pour la phase d'apprentissage, et la seconde partie (correspondant à 8294 images) pour le test. La figure 8.8 contient d'un côté les trajectoires ayant été utilisées pour l'apprentissage et de l'autre les trajectoires pour le test. Les tests réalisés ont permis d'obtenir une reconnaissance de phases de jeu de 76.1%. Ces résultats sont présentés à la figure 8.9. Ce taux peut être expliqué par le manque de données d'apprentissage. De plus, les principales erreurs observées sont les non-detections des phases d'arrêt de jeu (offensif ou défensif, phases numéro 3 et 8), le reste de l'interprétation du jeu étant assez proche de la réalité terrain.

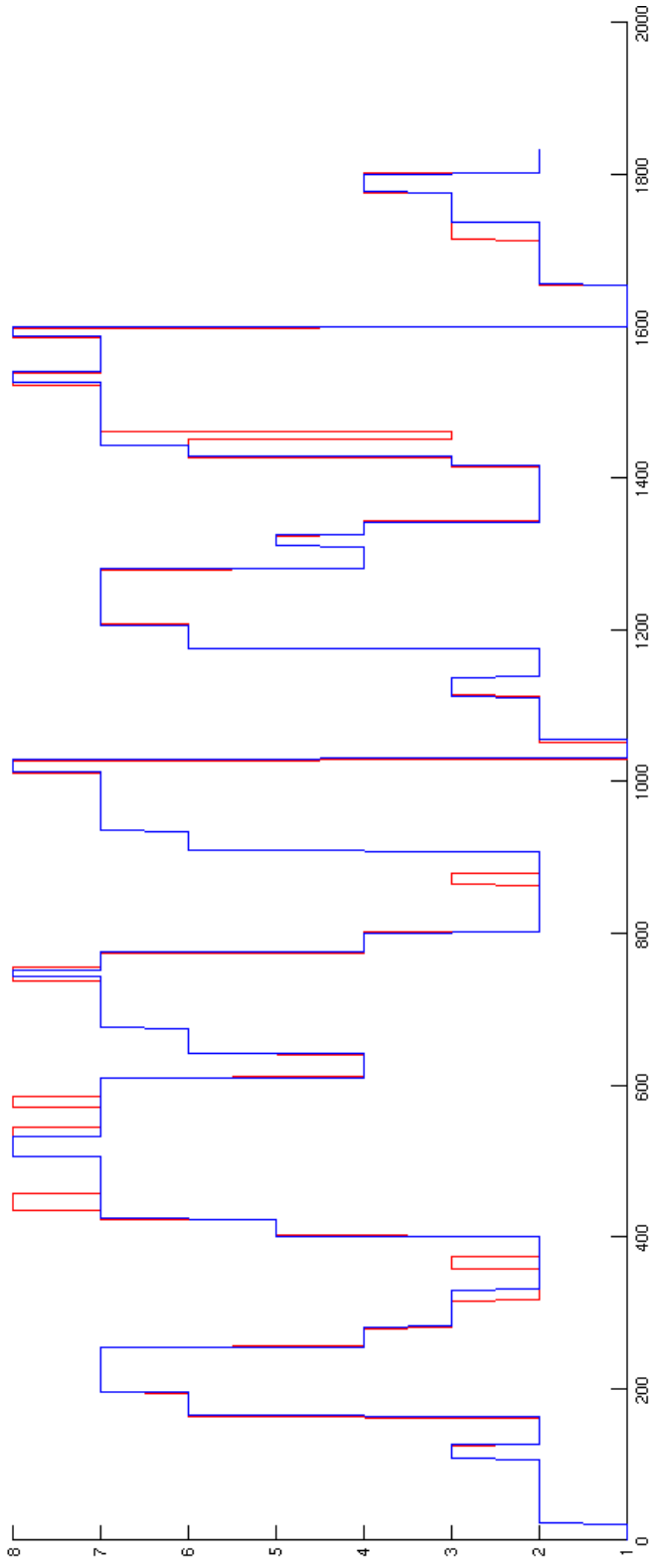

FIG. 8.7 – Présentation de l'interprétation obtenue par la méthode MCSM lorsque le même ensemble de trajectoires disponibles, issu de dix minutes de vidéos, est exploité dans l'apprentissage et dans le test. À la vérité-terrain tracée en rouge est superposée en bleu l'interprétation obtenue. En ordonnée, les nombres correspondent à la numérotation des phases de jeu décrite en introduction de la section 8.5.

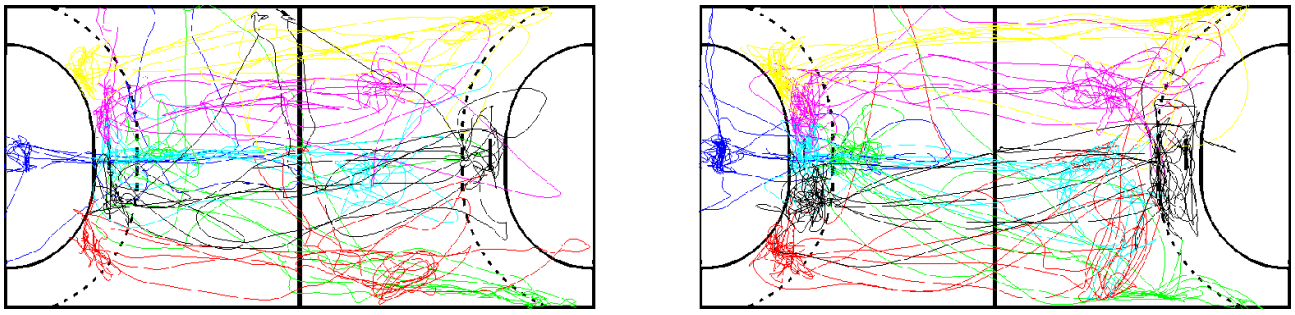

FIG. 8.8 – A gauche, les trajectoires ayant servi à l'apprentissage (trajectoires correspondant à 6370 images). A droite, les trajectoires traitées pour le test, correspondant à 8294 images.

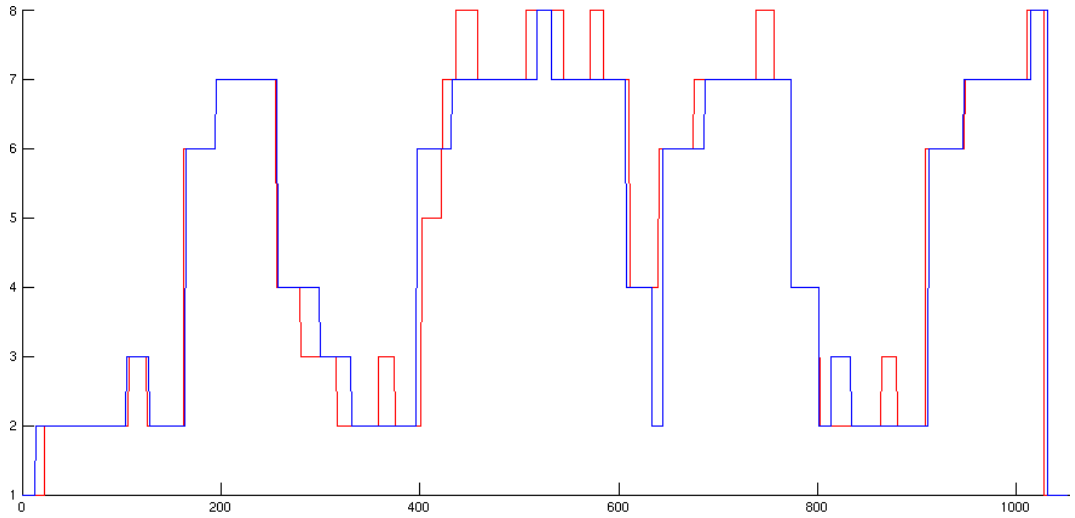

FIG. 8.9 – Présentation de l'interprétation obtenue par la méthode MCSM lorsque les trajectoires de la première partie de la vidéo (6370 images) sont considérées pour l'apprentissage et les trajectoires de la deuxième partie de la vidéo (8294 images) sont utilisées pour la phase de test. La vérité terrain est tracée en rouge, les résultats obtenus en bleu. En ordonnée, les nombres correspondent à la numérotation des phases de jeu décrite en introduction de la section 8.5.

8.5.1.1 Comparaison avec les résultats obtenus à l'aide des MMCH

Les résultats de reconnaissance de phases de jeu obtenus avec les MMCH diffèrent assez peu de ceux obtenus par les MCSM. En considérant l'ensemble des dix minutes de jeu tant pour l'apprentissage du modèle que pour la phase de test, un résultat d'interprétation correcte de 87.2% a été obtenu. Ce résultats doit être comparé avec le résultats de reconnaissance de phase de jeu de 89% obtenu par les MCSM.

De plus, lorsque l'on utilise la première partie des trajectoires disponibles (correspondant à 6370 images) pour l'apprentissage et la seconde partie des données (correspondant à 8294 images) pour le test, des résultats corrects d'interprétation de 72.7% ont été obtenus avec les MMCH contre les 76.1% obtenus avec les MCSM.

La comparaison entre la méthode MCSM et la méthode MMCH montre la pertinence de la modélisation MCSM. La méthode MCSM, qui prend en compte les temps de séjour dans les états de niveau supérieur, obtient de meilleurs résultats d'interprétation que la méthode MMCH.

Nous exploitons maintenant les données sonores contenues dans le flux sonore du match dont ont été extraites les trajectoires des joueurs. Nous effectuons une extraction automatique des coups de sifflet devant aider le système à mieux interpréter les phases de jeu observées. De plus, la prise en compte des coups de sifflet permettra d'élaborer une procédure de tests pouvant s'affranchir, dans une certaine mesure, du relatif faible volume de données disponibles pour l'apprentissage des modèles.

8.5.1.2 Intégration de données sonores : résultats de détection automatique des coups de sifflet

La méthode décrite en section 7.1.2 de détection des coups de sifflet a donné des résultats intéressants. Comme le montre la figure 8.11, en considérant comme requête un coup de sifflet issu d'un autre flux sonore, la méthode a permis de détecter 29 coups de sifflet sur les 31 coups contenus dans la vidéo, sans fausse alarme. Le seuil de détection a été fixé manuellement, ce qui nous a permis de choisir le seuil "optimal", c'est-à-dire permettant d'avoir le plus de bonnes détections, tout en ayant le moins de fausses alarmes. Une procédure de sélection automatique du seuil reste à proposer afin d'obtenir une extraction de coups de sifflet complètement automatique.

8.5.1.3 Intégration de données sonores : résultats d'interprétation de vidéos de handball

Nous pouvons maintenant utiliser les coups de sifflet (*i.e.*, les 29 coups de sifflet détectés) afin de tester les méthodes de prise en compte des coups de sifflet. Les coups de sifflet permettent d'aider la reconnaissance de phases de jeu dans un match de handball, chaque coup de sifflet correspondant à un changement de phase de jeu.

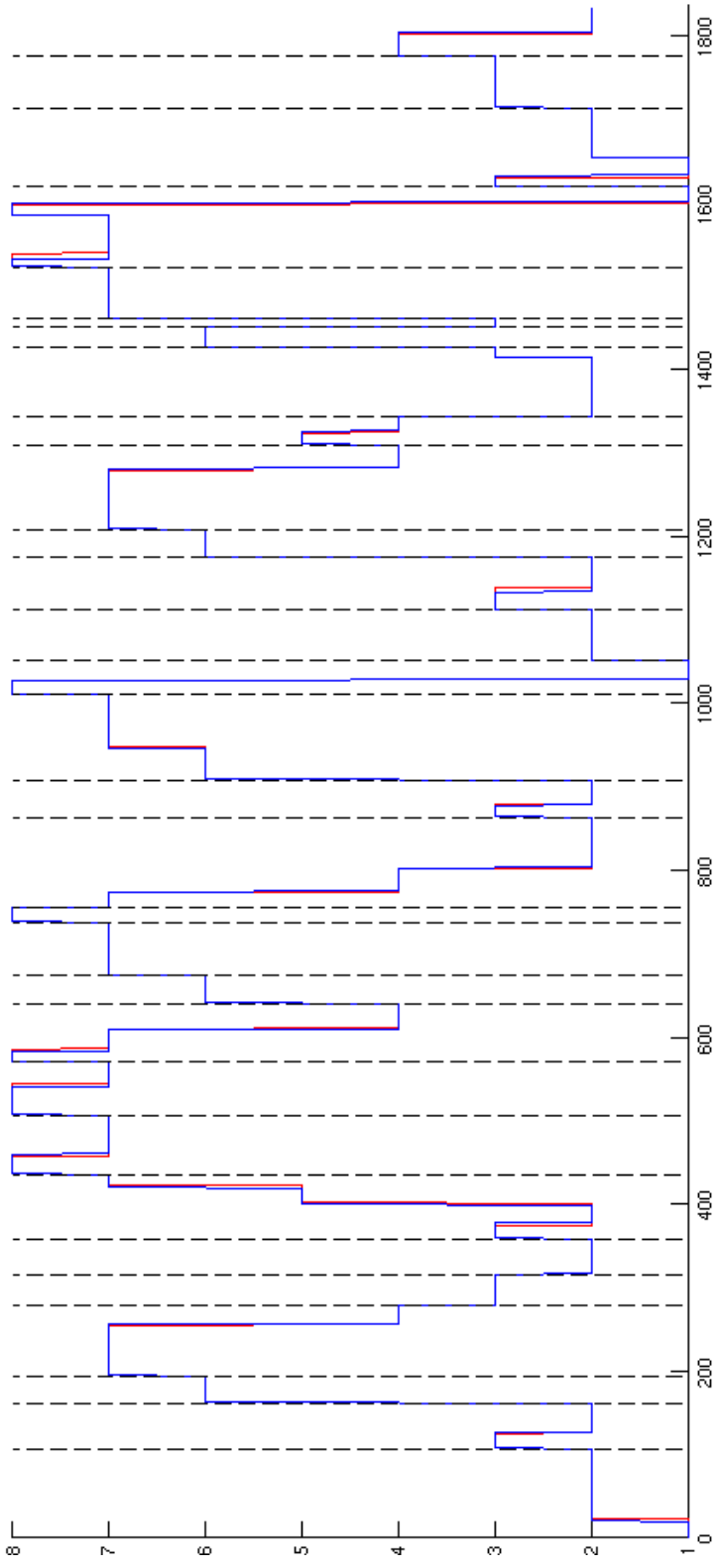

FIG. 8.10 – Présentation de l'interprétation obtenue par la méthode MCSM, avec prise en compte des coups de sifflets, lorsque le même ensemble de trajectoires disponibles, issu de dix minutes de vidéos, est exploitée dans l'apprentissage et dans le test. À la vérité-terrain tracée en rouge sont superposés en bleu les résultats obtenus. En ordonnées, les nombres correspondent à la numérotation des phases de jeu décrite en introduction de la section 8.5.

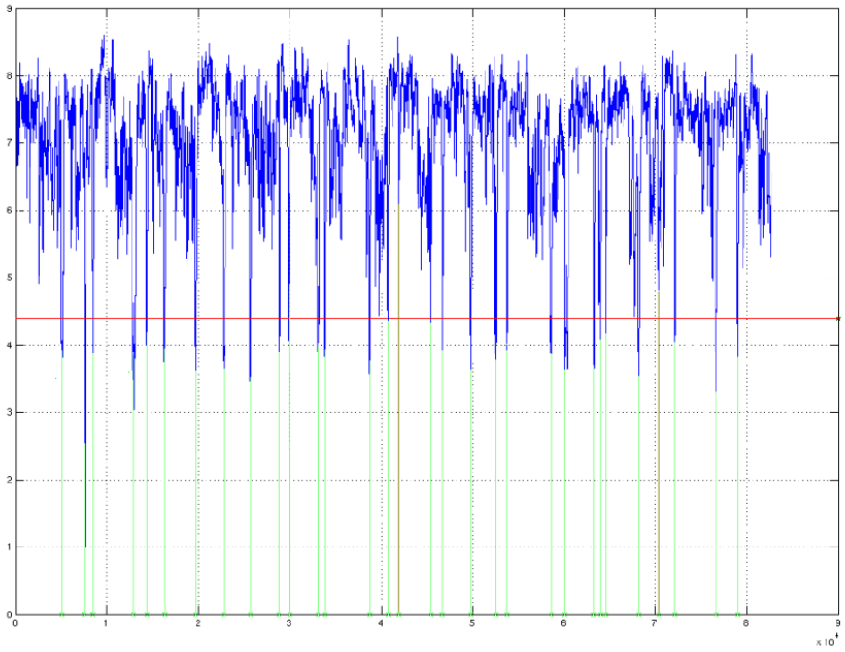

FIG. 8.11 – Présentation des résultats de détection des coups de sifflets obtenus sur les dix minutes de flux sonores. Le flux sonore est tracé en bleu, le seuil de détection est en rouge, les 29 coups de sifflet détectés sont repérés en vert alors que ceux non détectés (au nombre de deux) sont en marron.

En considérant l'ensemble des trajectoires de handball disponibles pour la phase d'apprentissage du modèle et pour la phase de test, le pourcentage de reconnaissance de phases de jeu obtenu par la méthode MCSM est de 97%. De plus, toute les phases de jeu sont correctement détectées, les 3% d'erreurs sont des imprécisions lors des transitions entre phases de jeu. La figure 8.10 illustre ce résultat.

Cette interprétation est à comparer au résultat de 89.2% obtenu sur les mêmes données, mais sans la prise en compte des coups de sifflet. Sans cette prise en compte des informations sonores (résultats présenté en figure 8.7), la méthode MCSM avait des difficultés à détecter les phases d'arrêt de jeu (phase numéro 3 et 8). Les résultats obtenus en exploitant les coups de sifflet montrent que cette information sonore permet de résoudre ce problème.

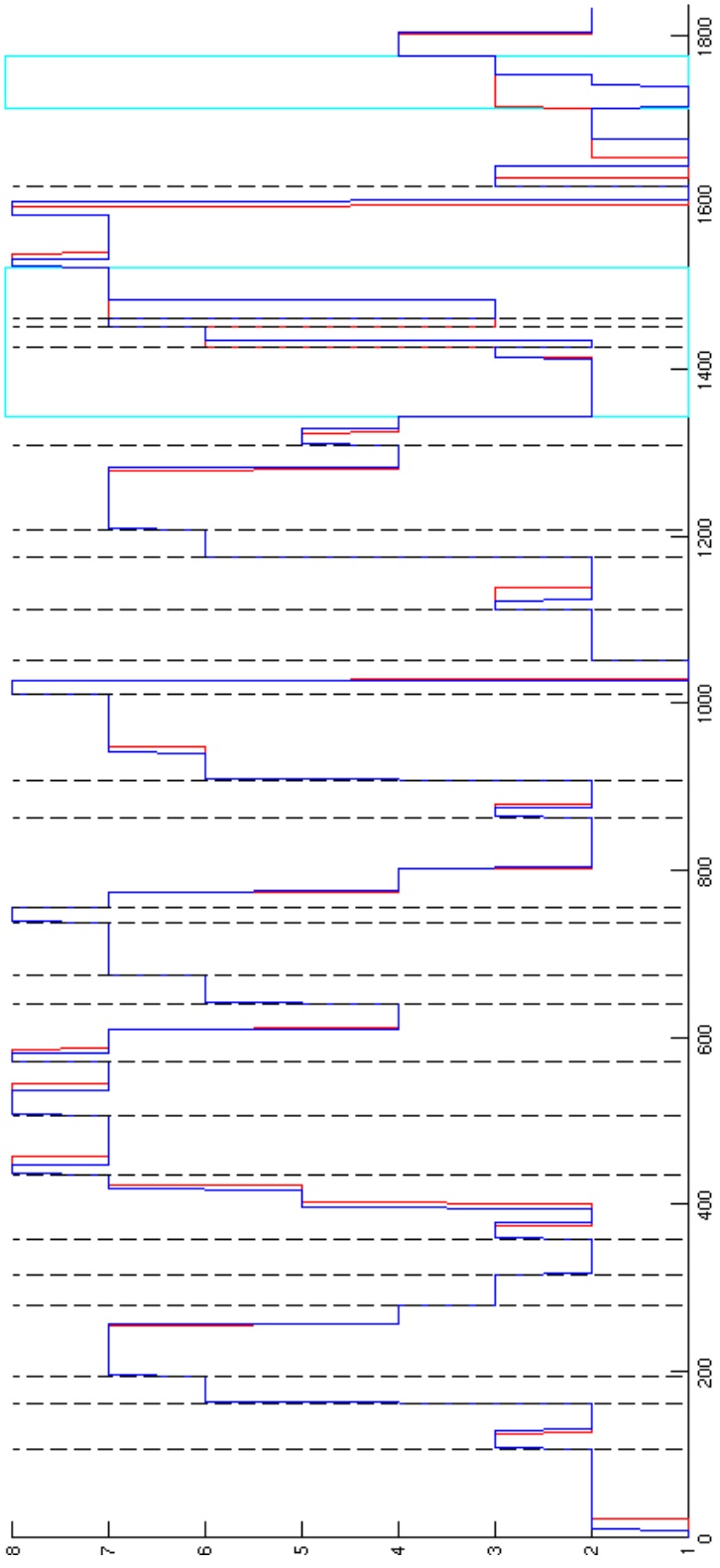

FIG. 8.12 – Présentation de l'interprétation obtenue par la méthode MCSM à l'aide de la validation croisée. À la vérité terrain tracée en rouge est superposée en bleu les résultats obtenus. Les coups de sifflet détectés sont indiqués en pointillés. Les évènements penalty et entre-deux sont entourés par un cadre bleu ciel. En ordonnée, les nombres correspondent à la numérotation des phases de jeu décrite en introduction de la section 8.5.

La prise en compte des coups de sifflet permet également de mettre en place une méthode de test par validation croisée. En effet, l'ensemble des dix minutes de trajectoires est "coupé" en 30 segments (délimités par les 29 coups de sifflets). Le décodage de Viterbi est alors effectué de façon individuelle pour chacun de ces segments (voir section 7.1.2.2), la seule information étant de changer de phase de jeu à chaque coup de sifflet. Ainsi, une méthode de tests par validation croisée a pu être exploitée, en effectuant le décodage de chacun des segments de façon indépendante des autres segments, ces derniers étant utilisés pour l'entraînement du modèle.

Ainsi, pour chaque segment de test, un ensemble de 29 segments (les autres segments) est retenu pour l'apprentissage du modèle. Cette procédure permet d'exploiter des ensembles de trajectoires d'apprentissage plus importants (correspondant à environ 9 minutes) sans avoir de juxtaposition entre les données d'apprentissage et de test.

La figure 8.12 présente la reconnaissance de phases de jeu obtenue, ainsi que les évènements "penalty" et "entre-deux" entourés d'un cadre bleu ciel. En utilisant cette méthode de validation croisée, la méthode MCSM obtient un résultat d'interprétation correcte de 89.8%.

Malgré ce résultat, on observe des erreurs de segmentation lors de segments de jeu correspondant à des évènements non observés dans les autres segments servant pour l'apprentissage. Ainsi, les cinq segments de jeu associés aux deux évènements "penalty" (et au retour de l'équipe dans sa moitié de terrain) et "entre-deux" (phase d'arrêt de jeu correspondant à un retour au milieu des deux équipes puis d'un entre-deux) ne sont pas ici finement interprétés comme la vérité terrain. En effet, les deux évènements "penalty" et "entre-deux" pourraient correspondre à des phases de jeu. Néanmoins, ces évènements sont observés une seule fois pendant les dix minutes de vidéos utilisées. Nous avons donc du utiliser les huit phases de jeu définies pour interpréter ces deux évènements, en collant au mieux aux activités observées. Avec une quantité de trajectoires suffisantes, ces deux évènements pourraient être modélisés par des états de niveau supérieur, c'est-à-dire par deux états de niveau supérieur S'_i correspondant aux phases de jeu "penalty" et "entre-deux". Cela illustre l'importance d'avoir un ensemble de données d'entraînement adapté à la modélisation développée.

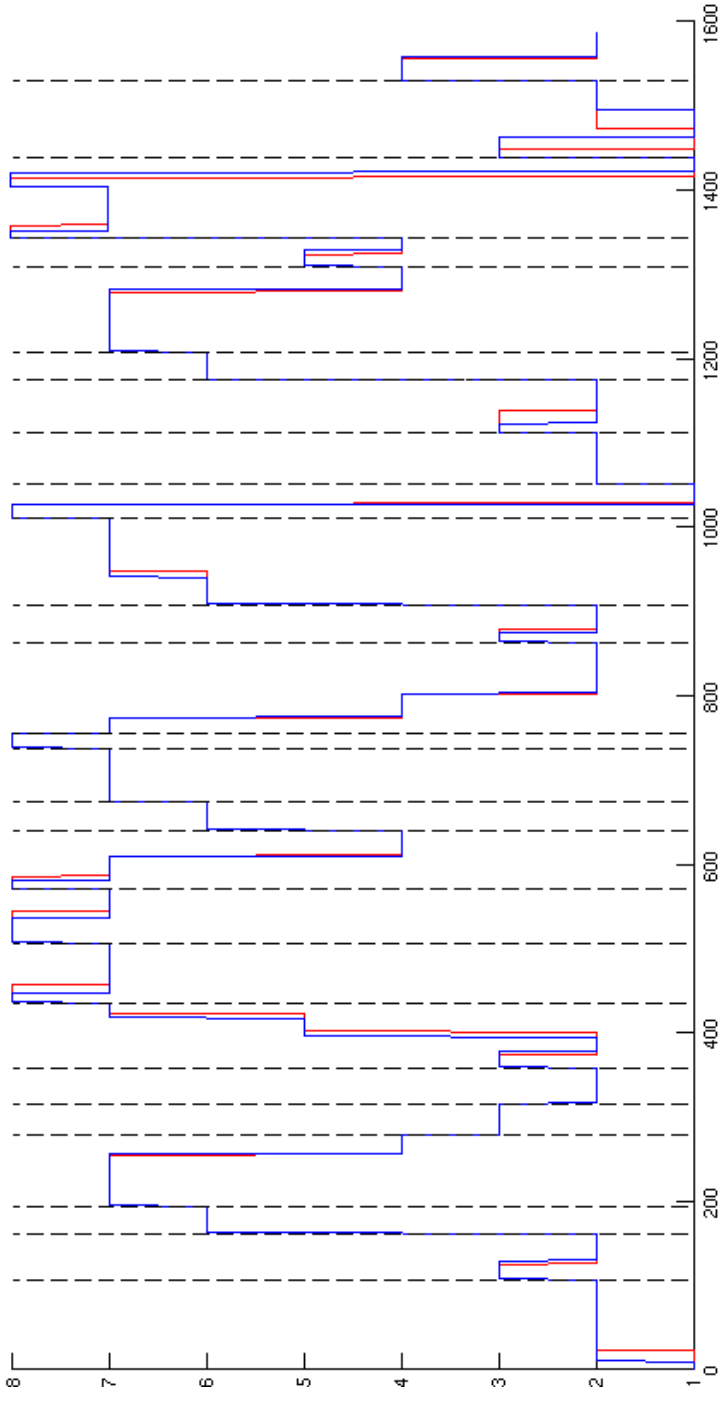

FIG. 8.13 – Présentation de l'interprétation obtenue par la méthode MCSM à l'aide de la validation croisée, sans considérer les segments correspondant aux évènements penalty et entre-deux. À la vérité terrain tracée en rouge est superposée en bleu l'interprétation obtenue. Les coups de sifflet sont détectés sont indiqués en pointillés. En ordonnée, les nombres correspondent à la numérotation des phases de jeu décrite en introduction de la section 8.5.

De même, la phase de retour au milieu du terrain après un but encaissé est observée seulement deux fois dans les dix minutes de jeu exploitées. Cette phase nécessiterait un état dédié, mais il n'a pas été possible d'efficacement entraîner un état de haut niveau de type "retour au milieu du terrain" avec les données disponibles. Nous avons donc segmenté, pour créer la vérité-terrain, les phases de retour au milieu du terrain après un but encaissé à l'aide des phases "arrêt de jeu défensif" et "montée lente en attaque". En effet, après un but encaissé, l'équipe à un mouvement comparable à celui d'un arrêt de jeu, avant de remonter au centre du terrain de manière identique que lors de montée lente en attaque.

La figure 8.13 et le tableau 8.1 présentent des résultats obtenus par la méthode MCSM en ne considérant pas les segments dans lesquels les évènements "penalty" et "entre-deux" sont observés. La figure 8.13 illustre l'interprétation alors effectuée. Le tableau 8.1 présente, pour chacune des huit phases de jeu définies, le taux de reconnaissance obtenu en comparaison à la vérité-terrain.

L'interprétation correcte présentée en figure 8.13 a donc été obtenue en ne considérant pas les cinq segments correspondant aux évènements "penalty" et "entre-deux", il reste alors environ 87% de la vidéo, c'est-à-dire 8 minutes et 30 secondes de contenu vidéo. Un résultat de 92.2% d'interprétation correcte a pu être obtenu par la méthode MCSM. Toutes les phases de jeu ont été correctement détectées, les 7.8% d'erreurs correspondant à des décalages entre la vérité-terrain et la reconnaissance de phases de jeu obtenue. Ces décalages, pour la plupart, correspondent à des décalages de l'ordre de une à deux secondes. Le résultat obtenu est donc une reconnaissance de phases de jeu satisfaisante de la vidéo observée (pour le niveau d'interprétation défini à l'aide des huit phases considérées).

8.5.1.4 Comparaison avec les interprétations obtenues à l'aide des MMCH utilisant les données sonores

L'utilisation des coups de sifflets par les MMCH se fait de manière identique à celle effectuée par les MCSM. Ainsi, chaque segment est décodé séparément, la méthode de test par validation croisée définie ci-dessus est donc également utilisable avec les MMCH.

Le résultat de reconnaissance de phases de jeu obtenu, en ne considérant pas les deux segments correspondant aux évènements "penalty" et "entre-deux", avec la méthode MMCH est de 89.8%. Ce résultat est à comparer avec le résultat de 92.2% obtenu avec la méthode MCSM. Pour la plupart des segments (plus précisément pour 28 des 30 segments exhibés à l'aide des 29 coups de sifflet détectés), les résultats obtenus avec la méthode MMCH sont identiques à ceux obtenus par la méthode MCSM.

	Taux de reconnaissance	Nombre d'images
Montée lente en attaque	68.1	728
Attaque placée	92.4	3776
Arrêt de jeu offensif	87.1	1056
Attaque rapide	95.6	1280
Retour rapide en défense	88.6	280
Retour lent en défense	93.6	1112
Défense placée	100	3592
Arrêt de jeu défensif	81	928

TAB. 8.1 – Présentation des taux de reconnaissance, en comparaison avec la vérité-terrain, obtenus par la méthode MCSM pour les huit phases de jeu définies. Ces résultats ont été obtenus à l'aide de la validation croisée sans considérer les segments correspondant aux évènements penalty et entre-deux. Le nombre d'images correspondant, dans la vérité-terrain, est également indiqué.

Néanmoins, pour certain segments, la méthode MMCH produit une reconnaissance de phases de jeu moins précise que la méthode MCSM. Ceci est illustré par la figure 8.14 qui présente l'interprétation obtenue par les méthodes MCSM et MMCH sur le premier segment de la vidéo de handball. L'interprétation obtenue sur ce premier segment par la méthode MMCH de 60%, contre 89% avec la méthode MCSM.

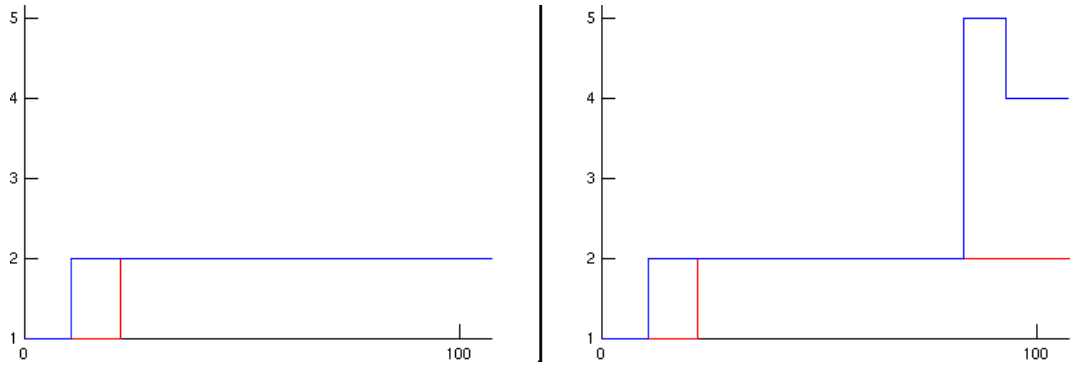

FIG. 8.14 – Présentation des résultats d'interprétation obtenus pour le premier segment de jeu. À gauche, les résultats obtenus à l'aide de la méthode MCSM ; à droite, ceux obtenus à l'aide de la méthode MMCH. À la vérité terrain tracée en rouge sont superposés en bleu les résultats obtenus. En ordonnée, les nombres correspondent à la numérotation des phases de jeu décrite en introduction de la section 8.5.

Ces résultats montrent que la prise en compte des temps de séjour effectuée par les MCSM permet d'obtenir des interprétations plus précises. D'une part, l'information de temps de séjour permet d'éviter des décalages temporels importants entre la vérité-terrain et l'interprétation obtenue lors des transitions entre phases de jeu. D'autre part, comme le montre la figure 8.14, cette information peut également permettre d'éviter des erreurs de reconnaissance de phases de jeu lors du décodage de Viterbi.

8.5.2 Conclusion

Nous avons présenté les résultats d'interprétation obtenus à l'aide des interactions entre joueurs de handball. La représentation proposée, à l'aide des interactions entre joueurs de handball et de leurs dynamiques a permis de caractériser les différentes phases de jeu. La méthode MCSM proposée dans le chapitre précédent a été testée et obtient des résultats de reconnaissance de phases de jeu satisfaisants en permettant une reconnaissance efficace des différentes phases de jeu. Ces résultats ont été obtenus avec un ensemble de trajectoires correspondant à seulement dix minutes de jeu. Des données supplémentaires pourraient permettre d'effectuer des interprétations plus précises, en considérant des phases de jeu supplémentaires et, ainsi, des scénarios plus divers.

8.6 Temps de calcul

Le temps de calcul nécessaire à l'ensemble de l'apprentissage pour la méthode MCSM, à l'aide des trajectoires des deux joueurs correspondant à 5 minutes de jeu, est d'environ 1 minute. Le temps nécessaire à la phase de test par la méthode MCSM permettant l'interprétation des trajectoires des deux joueurs correspondant à 5 minutes et 40 secondes de vidéo de squash est de 3 minutes. L'algorithme proposé permet la reconnaissance des phases de jeu seulement en post-traitement, c'est-à-dire une fois que la vidéo entière a été observée. Les temps de calcul des MMCH sont légèrement inférieurs. Ces temps de calcul ont été obtenus avec une valeur $k_{group} = 8$ et un PC standard. Les procédures les plus longues, pour le traitement de vidéos de squash, sont les calculs d'approximation des trajectoires nécessaires à l'obtention des descripteurs $\dot{\gamma}$.

Les temps de calcul présentés pour le handball correspondent aux tests réalisés à l'aide de la méthode MCSM avec validation croisée sur 8 minutes et 30 secondes, *i.e.*, en ne considérant pas les deux segments correspondant aux évènements "penalty" et "entre-deux" (le taux moyen d'interprétation obtenu correspondant étant 92.2%, le nombre de segments de jeu étant alors de 24). Le temps moyen calculé sur les 24 apprentissages réalisés par validation croisée (apprentissages sur des ensembles correspondant à 8 minutes de jeu) est de 3 secondes. Le temps de calcul nécessaire, par

exemple, à l'interprétation d'un ensemble de sept trajectoires correspondant à 35 secondes de vidéos est de 10 secondes. L'ensemble de l'interprétation des 8 minutes 30 secondes de jeu correspondant aux 24 segments est de 2 minutes et 20 secondes. Les temps de calcul des MMCH sont légèrement inférieurs. Ces temps de calcul ont été obtenus avec une valeur $k_{group} = 8$ et un PC standard. La différence observée avec les temps de calcul d'interprétation de vidéo de squash est expliquée par l'absence de calculs d'approximation des trajectoires nécessaires à l'obtention des descripteurs $\dot{\gamma}$.

De plus, la prise en compte des coups de sifflet permet d'interpréter chaque segment de jeu indépendamment. On peut donc, au fur et mesure de la vidéo, décoder les segments successifs observés. Dés qu'un coup de sifflet retentit, une dizaine de secondes, en moyenne, est nécessaire à la méthode MCSM pour interpréter les trajectoires correspondantes et donc le segment observé. La méthode MMCH, elle, permet d'interpréter les segments sans être obligé d'attendre la fin de ceux-ci. Des utilisations de la méthode MMCH pour l'interprétation en temps réel sont donc envisageables.

8.7 Conclusion

Des méthodes pour l'interprétation de vidéos de squash et de handball ont été testées. Des résultats intéressants, pour ces deux tâches de reconnaissance de contenu dans des vidéos de sports, ont été obtenus. Ces résultats mettent en valeur la pertinence des modélisations semi-markoviennes hiérarchiques parallèles mises en place. Les modélisations MCSM ont été utilisées pour l'interprétation de vidéos de sport et se comparent favorablement aux méthodes MMCH. Les représentations proposées prennent en compte les interactions entre les joueurs, à partir de leurs trajectoires, ainsi que leurs dynamiques. Elles permettent de caractériser les différentes phases de jeu observées dans des vidéos de squash et de handball. Les résultats obtenus permettent d'envisager l'extension de ces méthodes à d'autres sports individuels ou collectifs.

Conclusion

Dans cette partie, nous avons développé des méthodes statistiques pour la compréhension de phénomènes décrits par plusieurs trajectoires observées simultanément. Nous avons proposé des représentations permettant, pour différentes problématiques telles que la reconnaissance d'actions ou de phases de jeu dans des vidéos de sport, de prendre en compte les interactions entre les trajectoires vidéos observées. Les représentations définies, spécifiques aux tâches considérées, ont ensuite fait l'objet d'un traitement statistique permettant la reconnaissance de contenus dans des vidéos. Ainsi, des modélisations markoviennes hiérarchiques et parallèles ont été proposées pour prendre en compte la causalité temporelle des phénomènes observés pour la recherche automatique de contenus et pour l'interprétation de vidéos.

Les méthodes proposées pour la segmentation et la reconnaissance des phases de jeu de vidéos de sport (plus précisément le squash et le handball) ont fourni des résultats satisfaisants, offrant un outil de compréhension de haut niveau de vidéos de sport. Les expérimentations ont été effectuées sur des trajectoires de joueurs de squash obtenues à partir d'une caméra filmant une partie de squash du dessus et disponibles sur un serveur. Les trajectoires des joueurs de handball, également disponibles en ligne, ont été reconstruites, dans le plan du terrain de handball, à l'aide de deux caméras (chacune des caméras filmant respectivement une moitié du terrain de handball du dessus).

L'ensemble de ces premières validations expérimentales permet d'ouvrir un champ important de perspectives, concernant tant la prise en compte des interactions entre trajectoires dans des vidéos filmant des personnes (telles que la vidéo-surveillance) que pour l'analyse haut niveau automatique de vidéos de sport.

Conclusions et perspectives

Conclusion générale

"Je doute que toute la philosophie du monde parvienne à supprimer l'esclavage : on en changera tout au plus le nom. Je suis capable d'imaginer des formes de servitudes pires que les nôtres, parce que plus insidieuses : soit qu'on réussisse à transformer les hommes en machines stupides et satisfaites, qui se croient libres alors qu'elles sont asservies, soit qu'on développe chez eux, à l'exclusion des loisirs et des plaisirs humains, un goût du travail aussi forcené que la passion de la guerre chez les races barbares."

Marguerite Yourcenar - ***Mémoires d'Hadrien***

Dans ce travail de thèse, nous nous sommes intéressés à l'analyse de trajectoires issues de séquences d'images pour la reconnaissance de contenus dynamiques dans des vidéos. Dans cette conclusion générale, nous présentons une synthèse de nos travaux avant, dans la section suivante, de dégager quelques perspectives.

Synthèse des travaux effectués

Notre objectif a été de développer, pour chacune des tâches de reconnaissance abordées, des représentations adaptées des trajectoires vidéos traitées. Nous avons proposé des modélisations probabilistes pour prendre en compte les propriétés des trajectoires qui impliquent des procédures d'apprentissage statistique. Nous rappelons dans la suite nos contributions principales.

État de l'art de l'utilisation des séries temporelles et des trajectoires vidéos pour la reconnaissance de contenu

Un état de l'art (inexistant, à notre connaissance) des méthodes de traitement des séries temporelles, et plus particulièrement des trajectoires issues de vidéos, à des fins de reconnaissance de contenus a été proposé. Ce travail a permis de mettre en place le cadre de travail dans lequel nous nous sommes placés dans cette thèse, c'est-à-dire l'analyse des trajectoires (ainsi que des interactions entre trajectoires) observées dans des vidéos. La mise en place de ce contexte de travail a également permis de mettre en évidence les limitations et les difficultés rencontrées lors de tels travaux, et que nous avons tenté de résoudre.

Reconnaissance et détection de contenus vidéos à l'aide de trajectoires

Représentation invariante de trajectoires

Une représentation originale, avec des propriétés pertinentes d'invariance dans le plan image et permettant d'exploiter des trajectoires issues de vidéos acquises par des caméras mobiles, a été spécifiée. La caractérisation de trajectoires introduite, en plus de ses caractéristiques d'invariance, intègre les propriétés de dynamique (*i.e.*, l'évolution de la vitesse), et celles de forme (*i.e.*, l'évolution de la courbure). Ces propriétés permettent ainsi de traiter des trajectoires extraites de vidéos issues de caméra différentes. Une représentation continue des trajectoires étant nécessaire, une méthode de régression ainsi qu'une méthode de sélection du paramètre de lissage correspondant ont été définies.

Modèles de Markov cachés basés sur une quantification des données

La modélisation que nous avons développée s'appuie sur les modèles de Markov cachés. Les trajectoires extraites de plans vidéos étant généralement courtes, les modélisations classiques ne sont pas adaptées. Aussi, une modélisation originale (MMCQ) a été développée par modèles de Markov cachés qui repose sur une quantification des observations (ou MMCQ). Une distance entre trajectoires, reposant sur les MMCQ a également été développée.

Choix du nombre d'états des méthodes MMCQ

Une sélection automatique du nombre d'états de la quantification dans les MMCQ a été mise en oeuvre à partir d'outils statistiques.

Applications à des vidéos de sport

Nous avons testé la méthode MMCQ, et particulièrement la distance entre trajectoires, pour des tâches de classification, de clustering et de détection d'évènements inattendus. Des trajectoires extraites à l'aide de procédures de suivi dans des plans de vidéos de sport ont en premier lieu été utilisées. Les propriétés d'invariance de la représentation ainsi que sa propension à exprimer dynamique et forme ont ainsi été validées. L'efficacité des MMCQ, modélisant chaque trajectoire de façon individuelle, et de la distance entre trajectoires a également pu être démontrée. Des méthodes de classification et de clustering ont été mise en oeuvre et ont apporté des résultats convaincants. Enfin, les résultats de détection d'évènements inattendus mettent en évidence la capacité de notre cadre d'analyse pour prendre en compte la variabilité spatio-temporelle intra et inter-classes.

Reconnaissance de contenu vidéos à l'aide des interactions entre trajectoires

Nous donnons un bilan du traitement des interactions entre trajectoires issues de vidéos pour la reconnaissance de contenus.

Représentation des interactions entre trajectoires dans des vidéos de sports

Les représentations des interactions entre trajectoires ont été exploitées pour la caractérisation de phases de jeu dans des vidéos de sport, plus particulièrement pour le squash et le handball. Les trajectoires utilisées sont les trajectoires des joueurs. La représentation associée aux trajectoires de squash est composée de la caractérisation invariante de trajectoires proposée dans la deuxième partie de cette thèse, pour chacun des deux joueurs. De plus, la distance entre les joueurs permet de prendre en compte directement l'interaction entre les joueurs. La représentation développée permettant de caractériser les trajectoires des joueurs de handball d'une même équipe est composée de cinq distances. La position particulière du gardien de but a été prise en compte par la distance aux autres joueurs ainsi que les dynamiques des joueurs au travers de distances intra-trajectoires.

Modélisations markoviennes et semi-markoviennes hiérarchiques pour l'interprétation de vidéos de sport

Les représentations d'interactions de trajectoires ont fait l'objet d'un traitement statistique permettant la reconnaissance de contenus. Ainsi, des modélisations markoviennes et semi-markoviennes hiérarchiques et parallèles ont été proposées pour prendre en compte la causalité temporelle des phénomènes observés pour la recherche

automatique de contenus et pour l'interprétation (segmentation et reconnaissance simultanée de phase de jeu) de vidéos en phases de jeu. La représentation semi-markovienne permet de prendre en compte les temps de séjour dans les états de niveau supérieur correspondant aux phases de jeu, permettant une interprétation plus précise.

Applications du traitement des interactions entre trajectoires pour l'interprétation de vidéos de squash et de handball

Les méthodes proposées pour l'interprétation de vidéos de squash et de handball ont également fourni des résultats satisfaisants, offrant un outil de compréhension de haut niveau de vidéos de sport. Ces résultats, innovants en terme de compréhension de vidéos de sport, permettent de montrer la pertinence des représentations développées pour chacun de ces sports. De plus, les modélisations markoviennes et semi-markoviennes hiérarchiques parallèles semblent être un outil adapté pour la segmentation et la reconnaissance de vidéos de sport en phase d'intérêt. Enfin, la prise en compte de données sonores, par l'intermédiaire de détections de coups de sifflet, a permis d'améliorer sensiblement les résultats d'inteprétation des vidéos de handball.

Perspectives

"Actuellement, la science, dans le sens ancien du mot, a presque cessé d'exister dans l'Océania. [...] La méthode empirique de la pensée sur laquelle sont fondées toutes les réalisations du passé, est opposée aux principes les plus essentiels de l'Angsoc. Les progrès techniques eux-mêmes ne se produisent que lorsqu'ils peuvent, d'une façon quelconque, servir à diminuer la liberté humaine. [...]
Les deux buts du Parti sont de conquérir toute la surface de la terre et d'éteindre une fois pour toutes les possibilités d'une pensée indépendante. Il y a, en conséquence, deux grands problèmes que le Parti a la charge de résoudre : l'un est le moyen de découvrir, contre sa volonté, ce que pense un autre être humain, l'autre est le moyen de tuer plusieurs centaines de millions de gens en quelques secondes, sans qu'ils en soient avertis. Dans la mesure où continue la recherche scientifique, cela est son principal objet."

George Orwell - ***1984***

Nous présentons maintenant quelques perspectives à nos travaux.

Reconnaissance d'évènements vidéo à l'aide de trajectoires

Validation de la méthode pour des évènements correspondant à de plus grands ensembles de données

Les méthodes MMCQ d'analyse de trajectoires issues de vidéos acquises par des caméras mobiles devraient faire l'objet de tests supplémentaires. En effet, elles ont été testées sur un ensemble de trajectoires (issues de vidéos de ski et de Formule1) correspondant à 285 trajectoires au total. Des tests sur des données de plus grands ensembles, notamment pour la détection d'évènements inattendus, permettraient de

valider plus largement la méthode proposée. Les tests de détection d'évènements inattendus ont été effectués sur seulement cinq exemple. Une validation concluante de ces méthodes nécessiterait une quantité d'expérimentations plus importante.

Validation de la méthode de choix du nombre de bins dans des histogrammes pour des tâches de classification

La technique proposée en section 5.1.2 pour le choix du nombre d'états dans un MMCQ mériterait de faire l'objet d'une étude approfondie de son utilisation pour des histogrammes. Ainsi, il serait intéressant de comparer cette méthode avec d'autres méthodes existantes pour le choix du nombre de bins dans des histogrammes telle que celle proposée par Birgé et al. [Birgé 06].

Validation et extension de la méthode de reconnaissance de formes sur d'autres bases de données

Il serait intéressant de comparer la méthode de classification et de clustering de formes avec d'autres méthodes de classification invariante de formes telle que celle développée par Srivastava et al. [Srivastava 03].

Des procédures restent également à proposer pour le choix automatique du nombre de points d'échantillonnage, les résultats présentés dans ce document correspondant aux meilleurs résultats obtenus parmi différents échantillonnages. De plus, l'extension de la méthode à l'utilisation de chaînes semi-markoviennes telles que celles proposées dans la partie III pourrait permettre, en plus de la reconnaissance de formes, la segmentation des formes en parties d'intérêt. Par exemple, en plus de reconnaître une forme d'animal, il serait possible de segmenter la forme en parties, *i.e.*, reconnaître les bras, les jambes, la tête...

Reconnaissance de contenus vidéo par l'analyse des interactions entre trajectoires

Validation de la méthode d'interprétation de vidéos de handball et affinage des scénarios

Les résultats obtenus sur les trajectoires issues de vidéos de handball sont prometteurs. Malheureusement, comme précisé en section 8.5, les trajectoires correspondant à 10 minutes de vidéos ne suffisent pas à définir plus d'états de niveau supérieur que les huit états déjà considérés. Avec un ensemble d'apprentissage plus important, il serait probablement possible de définir des scénarios plus précis et des résultats d'analyse sémantique plus intéressants.

Extension des méthodes d'interprétation à d'autres sports individuels et collectifs

Les résultats obtenus pour le squash et le handball pourraient être aisément étendus à d'autres sport tels que le tennis, le football, le basket-ball ou tout autre sport se déroulant dans un espace clos. Le problème est d'obtenir ou de construire les données de trajectoires nécessaires à de telles extensions des modélisations semi-markoviennes. Certains sports, tels que le rugby, présenteraient néanmoins des problèmes d'occultations lors des procédures de suivi. En effet, lors des regroupements (mêlées fermées ou ouvertes), les joueurs seraient difficiles à identifier et à suivre précisément. Face à de telles problématiques, le recours à des techniques autres que la vision par ordinateur pourraient être exploitées, telle que l'utilisation de réseaux de capteurs permettant de suivre des cibles mobiles et ainsi de produire des trajectoires précises. On peut par exemple penser à des émetteurs placés dans les vêtements des sportifs qui permettraient d'éviter les difficultés rencontrées par les méthodes de suivi de vision par ordinateur.

Extension de la méthode à la vidéo surveillance à l'aide de modélisations de scénario issues du domaine de l'intelligence artificielle

Il serait intéressant d'essayer d'étendre les méthodes développées pour l'étude de vidéos de sport à l'analyse de contenus issus de vidéo surveillance. Ainsi, on peut penser que l'utilisation d'outils de scénarisation, à l'aide de grammaires logiques issues du domaine de l'intelligence artificielle telles que celles développées dans [Richardson 06], pourrait permettre de traiter la complexité et la diversité de contenus sémantiques observables en vidéo-surveillance.

Généralisation de la méthode pour l'utilisation simultanée du mouvement observé et des propriétés visuelles des objets suivis

Enfin, et toujours dans le but de pouvoir reconnaître des contenus sémantiques dans des vidéos diverses (et, notamment, dans le domaine de la vidéo-surveillance) et non plus seulement dans des vidéos de sport, il serait important d'effectuer une contextualisation automatique de l'étude de trajectoires formulée dans ce document en analysant les propriétés des objets suivis. En effet, dans ce document, les objets suivis, par hypothèse, sont connus, que ce soit les Formule1, les skieurs, les joueurs de squash ou de handball. Ainsi, une analyse contextuelle simultanée du mouvement des objets suivis ("comment ils bougent") en fonction de ce qui est observé ("qu'est-ce

qui bouge"), pourrait conduire à une analyse automatique des contenus vidéo plus intéressante et plus large.

Applications du traitement des interactions entre trajectoires en reconnaissance de gestes et d'actions

Des méthodes de classification et de reconnaissance de gestes humains, à l'aide des interactions entre trajectoires 3D de parties du corps humain, sont décrites dans la première partie des annexes. Les gestes sont définis, dans l'annexe A, comme des mouvements de courtes durées effectués avec la partie supérieure du corps.

Les méthodes développées demandent des validations expérimentales supplémentaires. Néanmoins, quelques résultats prometteurs ont déjà été obtenus, nous avons donc désiré les présenter dans ce document. Les méthodes développée permettent, même pour de très courts gestes, d'obtenir des résultats de classification et de clustering satisfaisants. Ces résultats reposent sur une représentation parallèle de la méthode MMCQ présentée en partie II.

Des méthodes pour la reconnaissance d'actions (définies, dans l'annexe A, par des mouvements de l'ensemble du corps, et de longueurs importantes) sont présentées. Une méthode d'extension de la méthode MCSM à l'interprétation de vidéos filmant plusieurs actions successives est également proposée.

Annexes

Annexe A

Reconnaissance de gestes et d'actions 3D à l'aide des interactions entre trajectoires

---

"Il faut voir la complexité là où elle semble en général absente comme, par exemple, la vie quotidienne. Cette complexité-là a été perçue et décrite par le roman du XIX^e siècle et du début du XX^e siècle.
Dans le même temps, au XIX^e siècle, la science a un idéal exactement contraire. Cet idéal s'affirme dans la vision du monde de Laplace, au début du XIX^e siècle. Les scientifiques, de Descartes à Newton, essayaient de concevoir un univers qui soit une machine déterministe parfaite. Mais Newton, comme Descartes, avait besoin de Dieu pour expliquer comment ce monde parfait était produit. Laplace élimine Dieu. Quand Napoléon lui demande « Mais monsieur de Laplace, que faites-vous de Dieu dans votre système ? », Laplace répond « Sire, je n'ai pas besoin de cette hypothèse ». Pour Laplace, le monde est une machine déterministe véritablement parfaite, qui se suffit à elle-même. Il suppose qu'un démon possédant une intelligence et des sens quasi infinis pourrait connaître tout évènement du passé et tout évènement du futur. En fait, cette conception qui croyait pouvoir se passer de Dieu avait introduit dans son monde les attributs de la divinité : la perfection, l'ordre absolu, l'immortalité et l'éternité. C'est ce monde qui va se détraquer puis se désintégrer."

Edgar Morin - ***Introduction à la pensée complexe***

La compréhension d'actions et de comportements dans des vidéos est actuellement d'un grand intérêt et constitue donc un pan important de la recherche en vision par

ordinateur [Laptev 07, Bobick 96, Yilmaz05, Piriou 06]. La problématique sous-jacente est l'exploitation sémantique des vidéos qui s'avère d'un intérêt certain pour des applications telles que la détection d'actions et d'activités en vidéo surveillance [Hu 07] ou l'indexation et la recherche de vidéos par le contenu, en particulier de vidéos de sport [Kokaram 06].

Dans de telles problématiques, les données fournies par des méthodes de suivi de certaines parties du corps humain, en termes de trajectoires, peuvent être d'une grande utilité pour la compréhension de gestes et d'actions à l'aide de vidéos.

De nombreux travaux ont traité le sujet de la reconnaissance d'actions dans des vidéos [Rao 02]. Parmi ceux-là, une attention particulière a été portée à la reconnaissance de mouvements humains dans le contexte vidéo, et plus spécifiquement la reconnaissance de gestes manuels [Davis 93, Yang 02, Bobick 97]. Les MMC ont été grandement utilisés pour modéliser les causalités temporelles contenues dans les gestes manuels (voir section 3.2.3). Siskind et al. [Siskind 96] ont, par exemple, présentés une méthode de classification de gestes par maximum de vraisemblance à l'aide de MMC, qui ne requiert pas la connaissance du type et de la pose de ces objets. Des travaux de Lee et al. [Lee 99] ont également exploités les MMC pour la reconnaissance de mouvements des mains, en considérant des informations sur les distributions de couleur dans les images (pour la procédure de suivi) ainsi que les directions des mouvements issus du suivi des mains pour la reconnaissance de gestes.

La reconnaissance de gestes et d'actions à partir d'informations 3D a été étudiée dans plusieurs travaux. Des méthodes récentes proposées par Weinland et al. [Weinland 07] ont mis en oeuvre une technique de reconnaissance d'actions vidéos utilisant une ou plusieurs caméras et reposant sur une connaissance a priori issues d'exemples 3D. Wilson et Bobick [Wilson 99] ont défini une nouvelle modélisation par MMC, les MMC paramétriques (MMCP), pour la reconnaissance de gestes. Ces MMCP ont l'avantage principal d'introduire un paramètre associé aux probabilités d'observation des MMC. Ils permettent de modéliser les variations d'amplitudes dans des gestes. Vogler et Metaxas [Vogler01] ont décrit une méthode basée sur les MMC Pa (voir section 1.3.4), appliqués à des données issues de tracking 3D pour la reconnaissance de mots dans le langage des signes. Pour cela, des représentations permettant de modéliser les mouvements des mains ainsi que d'autres informations telles que les notions d'ouverture et de fermeture des mains, informations importantes dans la compréhension du langage des signes ont été exploitées. Campbell et al. [Campbell 96] ont analysé les invariances des représentations pour la reconnaissance de gestes 3D.

Des méthodes pour le suivi de parties du corps humain dans des vidéos ont ainsi été développées notamment par Fourès et Joly [Fourès 03a, Fourès 03b] qui utilisent un modèle décrivant les mouvements humains à l'aide d'une décomposition du corps

humains en rubans et en sous-rubans. Une décomposition en niveau hiérarchique des mouvements à l'aide modèle graphiques est effectuée, les données d'un premier niveau de décomposition [Fourès 03a] étant utilisées et affinées afin d'effectuer un suivi efficace des différentes parties du corps [Fourès 03b]. D'autre travaux, par Bernier et al. [Bernier01], inspiré des travaux de Wren et al. [Wren 97], effectue une reconstruction des trajectoires 3D à l'aide de deux caméras et d'une procédure de suivi de blob 3D.

Quelques techniques ont été récemment proposées pour la reconnaissance de gestes à l'aide des seules données de trajectoires 3D. Des travaux, par Marcel et al. [Marcel 00, Just 04] ont utilisé des MMC E/S (voir section 1.3.2) afin de modéliser les coordonnées 3D et leurs variations. Pour cet objectif, ils ont mis à disposition une base de données de trajectoires 3D de gestes humains ([Interactplay]). Des dictionnaires de primitives d'actions, correspondant à des segments de trajectoires 3D, sont introduits dans [Raptis 08] pour la reconnaissance d'actions. La reconnaissance (supervisée) est effectuée par une méthode de type "bag of features" avec classification par séparateurs à vastes marges (SVM). Ali et al. [Ali 07] ont eu recours à des trajectoires d'éléments du corps humain, aussi bien 2D que 3D pour la reconnaissance d'actions. Ils ont fait appel à la théorie des systèmes chaotiques afin de modéliser les dynamiques non linéaires observées dans les actions.

Notre objectif est tout d'abord de concevoir une méthode de reconnaissance de gestes manuels prenant en compte les interactions entre les différentes parties du corps. Nous pourrons ainsi montrer l'importante information sémantique contenue dans les interactions entre trajectoires pour des tâches de reconnaissance de gestes. Ensuite, cette méthode sera étendue aux actions mettant en jeu l'ensemble du corps. Nous visons une méthode indépendante des personnes effectuant les actions, c'est-à-dire devant être invariante aux différences de taille pouvant exister entre les acteurs, ainsi qu'aux variations d'amplitude pouvant apparaître dans une même classe de gestes et d'actions. Enfin, les gestes considérés (au contraire des actions) étant relativement courts en temps d'exécution, les séquences d'images correspondantes (et donc les trajectoires correspondantes) peuvent être de durée très réduite. La méthode proposée devra donc être capable de traiter efficacement des ensembles de données de "faibles" tailles, pour, par exemple, des tâches de clustering de gestes. Enfin, tous les paramètres de la méthode devront pouvoir être automatiquement fixés.

Dans la section suivante, nous introduisons la méthode développée pour la reconnaissance de gestes 3D à l'aide des trajectoires de différentes parties du corps.

A.1 Reconnaissance de gestes 3D "courts" à l'aide des interactions entre parties du corps humain

Les gestes, au contraire des actions qui seront traitées dans la section suivante, correspondent à des mouvements généralement assez courts effectués avec la partie supérieure du corps. Les gestes traités sont des gestes réalisés par des acteurs filmés par plusieurs caméras. Le mouvement 3D de certaines parties du corps (les mains, la tête ainsi que le torse) ont été reconstruits. Nous avons ainsi à disposition les trajectoires 3D pour la reconnaissance, supervisée ou non, de gestes. Les gestes qui peuvent correspondre à des mouvements de nages ou d'applaudissements par exemple, sont de durée courtes, appréhendés en moyenne par quelques dizaines d'observations (une par image) seulement.

Nous allons tout d'abord décrire la représentation des gestes que nous avons définies. Elle prend en compte les interactions entre les trajectoires 3D observées. Ensuite, nous décrirons l'utilisation faite des MMCQ (voir section 5.1.1), selon une approche "parallèle". Ils seront dénommés MMCQ Pa. Nous les avons exploités pour la reconnaissance et le clustering de gestes manuels "courts". Les trajectoires qui sont prises en compte dans cette section correspondent aux coordonnées successives des centres des éléments du corps humain suivis et reconstruits en 3D à l'aide de plusieurs caméras (voir plus précisément dans les expérimentations exposées dans le chapitre suivant).

A.1.1 Représentation de gestes 3D à l'aide d'interactions entre parties du corps humain

Nous introduisons dans ce paragraphe la représentation retenue des interactions entre les trajectoires exprimant les gestes étudiés. Le mouvement d'une partie du corps humain (*i.e.*, les mains, le torse ou la tête) est décrit par une trajectoire 3D correspondant aux positions successives de cette entité dans l'espace. Pour une vidéo contenant n images, nous avons la trajectoire $T = \{(x_1, y_1, z_1), .., (x_n, y_n, z_n)\}$.

Afin d'obtenir la représentation désirée, cinq éléments de caractérisation sont considérées (voir figure A.1). Il s'agit de :

- la distance entre les deux mains, notée d_{GD},

- les distances entre la tête et respectivement les mains gauche et droite et la tête, notées d_{TG} et d_{TD},

- les angles entre le corps et respectivement les bras gauche et droit, notés θ_{CG} et θ_{CD}.

Ces cinq variables forment une représentation adaptée, en termes de distances et d'angles, de gestes 3D. Elles exploitent les interactions entre les trajectoires des quatre membres.

Les distances introduites sont formulées par les distances euclidiennes en 3D, *i.e.* la distance entre deux points A et B, de coordonnées respectives $\{x_A, y_A, z_A\}$ et $\{x_B, y_B, z_B\}$:

$$d_{AB} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2}.$$

Les angles utilisés dans cette représentation entre le corps (dont la direction est donnée par le vecteur $\overrightarrow{TB}$) et les mains gauche et droite, de direction respective $\overrightarrow{TG}$ et $\overrightarrow{TD}$, notés respectivement θ_{CG} et θ_{CD}, sont donnés par :

$$\theta_{CG} = \arccos(\frac{\overrightarrow{TB}.\overrightarrow{TG}}{|TB|.|TG|}) \text{ et } \theta_{CD} = \arccos(\frac{\overrightarrow{TB}.\overrightarrow{TD}}{|TB|.|TD|})$$

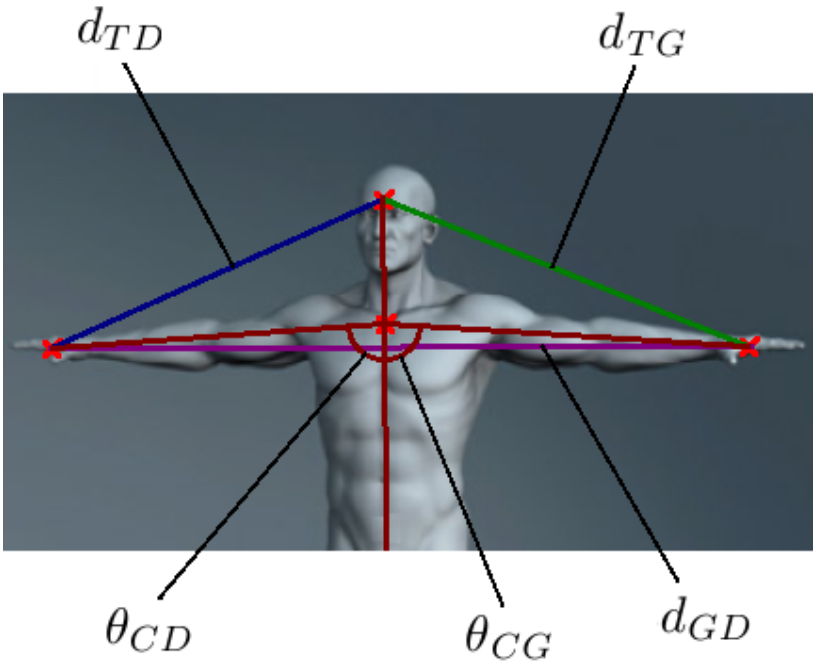

FIG. A.1 – Les cinq variables considérées pour la représentation de gestes à l'aide des interactions entre les mains, le torse et la tête.

Finalement, les gestes seront représentés par les valeurs centrées et normalisées des cinq variables présentées ci-dessus, *i.e.*,

$$\tilde{d}_{GD} = \frac{d_{GD} - d_{GD}^{mean}}{d_{GD}^{mean} + C}, \tilde{d}_{TG} = \frac{d_{TG} - d_{TG}^{mean}}{d_{TG}^{mean} + C}, \tilde{d}_{TD} = \frac{d_{TD} - d_{TD}^{mean}}{d_{TD}^{mean} + C}$$

et

$$\tilde{\theta}_{CG} = \frac{\theta_{CG} - \theta_{CG}^{mean}}{\theta_{CG}^{mean} + C}, \tilde{\theta}_{CD} = \frac{\theta_{CD} - \theta_{CD}^{mean}}{\theta_{CD}^{mean} + C}$$

où les valeurs moyennes sont calculées pour chacun des gestes sur leur durée, et C une constante permettant d'éviter des valeurs nulles aux dénominateur.

Cette modélisation permet d'obtenir l'invariance désirée aux possibles variabilités d'amplitude des gestes (à l'aide des opérations de centrage et de normalisation). Cette variabilité dans les amplitudes des gestes observés apparaît du fait de la diversité d'effectuer un geste (deux acteurs effectuant un geste de nage ne le ferons pas nécessairement avec la même amplitude), ainsi que des différences physiologiques (*i.e.*, les différences de taille) entre acteurs. De plus, cette représentation est directement invariante aux transformations, dans l'espace, de translations, de rotations (propriétés des distances relatives ainsi que des angles relatifs), sans nécessiter de recalage dans l'espace.

A notre connaissance, une seule méthode a été proposée pour la reconnaissance de gestes manuels à partir du suivi 3D des mains, de la tête et du torse, par Just et al. [Just 04], dans laquelle les vecteurs de représentation sont composés par, notamment, les valeurs des coordonnées spatiales des mains. Ainsi, cette représentation n'est pas invariante aux transformations spatiales considérées. Pour comparer des gestes en utilisant une telle méthode, il est alors requis que les acteurs se trouvent placés de la même façon, au même endroit, avec la même orientation (le facteur d'échelle restant alors un problème quant aux différences de tailles des acteurs). Ces questions peuvent néanmoins être gérées par des recalages préalables. Cependant, le recalage en rotation ne paraît pas aisé, alors que notre représentation permet de satisfaire directement aux invariances souhaitées.

Les vecteurs de caractérisation des gestes sont donc les cinq vecteurs contenant les valeurs successives des cinq variables. Par exemple, le vecteur de caractérisation correspondant à la valeur $\tilde{d}_{GD}$ pour un geste G_k (geste d'index k composé de n_k images) sera :

$$V^k_{GD} = [\tilde{d}^k_{GD,1}, \tilde{d}^k_{GD,2}, ..., \tilde{d}^k_{GD,n_k-1}, \tilde{d}^k_{GD,n_k}].$$

A.1.2 Modélisation et comparaison de gestes 3D par MMCQ Pa

Nous allons exploiter les MMCQ, et plus particulièrement les MMCQ parallèles pour la modélisation et la reconnaissance de gestes. En effet, les gestes sont caractérisés par un faible nombre d'observations. Dans les expérimentations faites dans le chapitre suivant, la taille moyenne des trajectoires associées à chaque geste est de 25 valeurs. Nous pourrons ensuite les comparer aux MMC/MMG Pa afin de mettre en valeur les propriétés des MMCQ Pa pour le traitement d'ensembles réduits de données.

A.1.2.1 Modélisation de gestes 3D par MMCQ Pa

Dans cette section, nous proposons une extension des MMCQ, appelée MMCQ Pa pour MMCQ parallèles. Elle correspond à l'adaptation de la modélisation MMC Pa au MMCQ. La modélisation MMC Pa a été proposée et développée par Bourlard [Bourlard 97] pour la reconnaissance de parole puis utilisée notamment par Vogler et Metaxas ([Vogler01] pour la reconnaissance du langage des signes dans des vidéos. L'idée, simple, est de modéliser chaque source d'observations associée à une action par une modélisation indépendante. En effet, plutôt que de modéliser les observations de toutes les sources en un schéma unique, comme peuvent le faire les MMC factoriels [Ghahramani 97], modélisations dont les procédures d'apprentissage de type *EM* peuvent s'avérer très coûteuse, l'idée est ici de poser une hypothèse d'indépendance entre sources d'observations afin de modéliser chacune de ces sources séparément (voir section 1.3.4).

L'hypothèse faite pour les MMCQ Pa est simplement que l'ensemble des signaux considérés évoluent de façon synchrone et indépendante. Ainsi, considérons les calculs de maximum de vraisemblance d'un MMCQ Pa,

$$\max_{Q^1,\ldots,Q^C} \{\log P(Q^1,\ldots,Q^C,O^1,\ldots,O^C|\lambda_1,\ldots,\lambda_C)\},$$

où Q^i est la séquence d'état associée au signal i, ayant pour séquence d'observations est O^i et le MMCQ associé est décrit par λ_i. L'hypothèse d'indépendance des processus donne

$$\max_{Q^1,\ldots,Q^C} \{\log P(Q^1,\ldots,Q^C,O^1,\ldots,O^C|\lambda_1,\ldots,\lambda_C)\} = \max_{Q^1,\ldots,Q^C} \left\{ \sum_{i=1}^{C} \log P(Q^i,O^i|\lambda_i) \right\}.$$

Ainsi, nous avons recours à un MMCQ Pa permettant de modéliser chacune des cinq sources d'observations, *i.e.*, les vecteurs de caractérisation de gestes V^k_{GD}, V^k_{TG}, V^k_{TD}, V^k_{CD} et V^k_{CD}.

A.1.2.2 Comparaison de gestes 3D par MMCQ Pa

Afin de comparer deux gestes, une mesure de similarité doit être définie entre les MMCQ Pa introduits précédemment. Pour cela, la distance croisée D_c entre MMCQ définie en section 5.1.3 est ici retenue. La distance permettant la comparaison de gestes est alors, suite à l'hypothèse d'indépendance entre les cinq vecteurs de caractérisation, la somme des distances D_c entre MMCQ respectifs composant les MMCQ Pa. Nous pouvons ainsi comparer deux gestes G_i et G_j par la distance définie ci-dessous :

$$D_{gest}(G_i,G_j) = \sum_{l=1}^{5} D_c^l(\lambda_i^l,\lambda_j^l),$$

où l correspond à l'index décrivant les cinq variables utilisées pour la représentation de gestes (*i.e.*, $\tilde{d}_{GD}$, $\tilde{d}_{TG}$, $\tilde{d}_{TD}$, $\tilde{\theta}_{CG}$ et $\tilde{\theta}_{CD}$), les λ_i^l correspondent aux paramètres (A, B, Π) des MMCQ associés à ces mêmes variables pour le geste i.

A.1.2.3 Choix du nombre d'état des MMCQ Pa

Le choix automatique du nombre d'état exploitée pour les MMCQ est également utilisé pour les MMCQ Pa. La méthode de sélection du nombre d'état décrite en section 5.1.2 est ainsi appliquée à tout les MMCQ composant l'ensemble des MMCQ Pa. Un nombre d'état unique est donc considéré pour tout les MMCQ respectifs composant les MMCQ Pa.

A.1.3 Reconnaissance de gestes 3D par MMCPa

Nous présentons maintenant les tâches de reconnaissance de gestes mises en place. Celles-ci s'appuie sur la distances entre gestes D_{gest}.

A.1.3.1 Reconnaissance supervisée de gestes 3D par MMCQ Pa

La reconnaissance supervisée de gestes 3D à l'aide des trajectoires de parties du corps humain est menée de manière similaire à la reconnaissance d'évènements dans des vidéos présentée en section 5.2.1.1, en utilisant les MMCQ Pa en lieu et place des MMCQ. Chaque classe de gestes est représentée par l'ensemble des MMCQ Pa associés aux gestes utilisés pour l'apprentissage de cette classe. La reconnaissance de gestes s'effectue à l'aide d'une méthode d'agrégation par lien moyen, *i.e.*, en calculant la moyenne des distances entre le geste G_k et toutes les gestes G_l de la classe C_i par :

$$D_{lm}(G_k, C_i) = \frac{\sum_{G_l \in C_i} Dist(G_k, G_l)}{\#C_i}.$$

A.1.3.2 Clustering de gestes 3D par MMCQ Pa

Le clustering de gestes est effectué par une méthode de classification ascendante hiérarchique. Chaque geste représentant initialement une classe, la distance entre deux classes C_i et C_j de gestes est définie, à l'aide d'une méthode d'agrégation par lien moyen, par :

$$D_{lm}(C_i, C_j) = \frac{\sum_{G_k \in C_i, G_l \in C_j} Dist(G_k, G_l)}{\#C_i \#C_j}.$$

A.1.4 Tests effectués et résultats obtenus

Des bases de données, disponibles en ligne, de gestes [Interactplay] et d'actions [MoCapDATA] ont été utilisées pour tester les méthodes proposées de reconnaissance

de mouvement à partir des interactions entre trajectoires de différentes parties du corps. Il s'agit de classification et de clustering de gestes et d'actions 3D.

À notre connaissance, une seule méthode a été jusqu'à présent proposée [Just 04] pour la reconnaissance de gestes à partir des trajectoires 3D de membres du corps humain. La méthode développée dans [Just 04] s'appuie sur des descripteurs comprenant les coordonnées spatiales des membres suivis. Elle requiert que les actions soient effectuées aux mêmes positions et orientations dans le repère 3D. Afin d'avoir une méthode invariante aux translations et aux rotations dans le repère 3D, une opération de recalage doit être effectuée. Cette dernière peut conduire à des imprécisions pour les tâches de reconnaissance considérées. La méthode que nous avons définie en section A, utilisant les interactions entre trajectoires, permet d'assurer directement l'invariance aux translations et aux rotations.

De plus, dans la méthode proposée dans [Just 04], les trajectoires des personnes gauchères ont été symétriquement transformées afin d'avoir les caractéristiques de personnes droitières. Notre méthode traite chaque geste de façon indépendante, permettant ainsi, comme nous le verrons dans la section A.1.4.2, de détecter les mouvements de gaucher comme formant à une classe particulière de "gaucher effectuant tel geste" ou bien (selon la définition des classes d'apprentissage) de reconnaître des gestes donnés, qu'ils soient effectués par un gaucher ou un droitier. La méthode [Just 04], dans laquelle un ensemble de gestes doit être utilisé pour l'entraînement des MMC E/S associés aux classes de gestes, ne peut appréhender qu'une tâche de classification. Par contre, notre méthode peut s'appliquer à une reconnaissance non-supervisée (ou clustering) d'un ensemble de gestes (voir section A.1.4.2).

A.1.4.1 Description des données utilisées

Afin de tester notre méthode pour la reconnaissance de gestes 3D, nous avons utilisé une base de données ([Interactplay]) de trajectoires 3D reconstruites dans des vidéos représentant des acteurs qui effectuent des gestes. Les trajectoires 3D ont été reconstruites à l'aide de deux caméras et d'une procédure de suivi de régions 3D, de formes ellipsoïdes (les détails de la méthode de suivi de régions 3D peuvent être trouvés dans [Bernier01]). Les trajectoires mises à disposition sont extraites du suivi en considérant les centres de ces ellipsoïdes.

Pour tester la méthode proposée, nous avons exploité six classes de trajectoires 3D fournies par la base de données. Ces six classes de gestes sont les suivantes :

- mouvement de nage (illustrée dans la figure A.2),
- mouvement de vol, *i.e.*, de battements d'ailes,
- applaudir,
- pointer du doigt,

- mouvement, de la main, de négation,
- mouvement, de la main, d'au revoir.

Il est à noter que trois des six classes correspondent à des gestes mono-manuels (*i.e.*, les trois dernières classes parmi les six présentées) alors que les trois autres classes sont des gestes effectués avec les deux mains.

La figure A.2 présente un ensemble d'images extraites d'une vidéo filmant un acteur effectuant un geste de la classe "nage". Comme on peut le voir sur ces images, les acteurs portent des gants de différentes couleurs permettant d'aider la procédure de suivi et de reconstruction 3D.

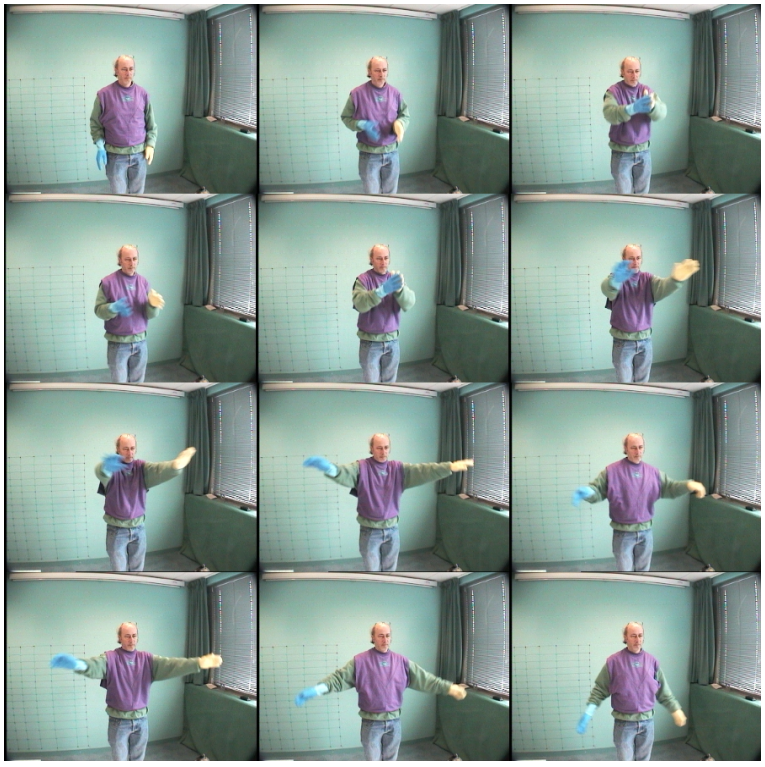

FIG. A.2 – Images appartenant à un geste de "nage" (à lire de gauche à droite et de haut en bas).

Pour chacune de ces classes de gestes, la base de données fournit les trajectoires reconstruites en 3D des mains, de la tête ainsi que du torse que nous avons exploitées pour tester les méthodes de classification et de clustering proposées en section A.1. Pour chaque classe de gestes mise à disposition, une vingtaine d'acteurs ont effectué les actions, lors de 5 session différentes. À chacune de ces sessions, les acteurs ont réa-

lisé 10 fois le même geste de sorte que la base de données met à disposition environ 1000 ensembles de 5 trajectoires (un ensemble par geste) par classes de geste.

La méthode que nous avons proposée en section A.1, au contraire de [Just 04] qui entraîne un MMC E/S par classe de gestes, compare les gestes un à un. Cela est en effet nécessaire afin d'envisager les tâches de clustering de gestes. Néanmoins, les gestes étant comparés de façon individuelle, il est important afin de ne pas introduire de biais dans les résultats présentés, de considérer un seul geste par acteur. En effet, nous souhaitons mettre en valeur les propriétés d'invariance à la diversité possible dans la réalisation un geste. Ainsi, comme nous l'avons souligné dans la section A, nous visons une méthode indépendante des personnes effectuant les actions, c'est-à-dire devant être invariante aux différences de taille pouvant exister entre les acteurs ainsi qu'aux variations d'amplitude pouvant apparaître dans une même classe de gestes.

Nous avons, pour ces raisons, construit deux ensembles distincts de gestes qui seront utilisés dans la suite de cette section, chacun composé de 120 actions. Chacun de ces deux ensembles de gestes contient un nombre égal (*i.e.*, 20 gestes) de gestes pour chacune des six classe d'actions présentées ci-dessus. Dans chaque classe de chaque ensemble de trajectoires considérées, chaque action a été exécutée par un acteur différent.

Deux ensembles de tests ont été construits afin d'avoir une validation des méthodes de reconnaissance de gestes. Les méthodes proposées ont été testées séparément sur chacun de ces deux ensembles. Les résultats de classification et de clustering correspondent aux moyennes des résultats obtenus sur ces deux ensembles.

Il est enfin important de souligner que les séquences de coordonnées 3D exploitées sont de tailles réduites. En effet, la moyenne de la taille des trajectoires exploitées sur les 240 gestes (1200 trajectoires) et de 25 points.

A.1.4.2 Résultats de reconnaissance de gestes 3D par MMCQ Pa

Les résultats de la méthode de classification et de clustering de gestes 3D par MMCQ Pa développée sen section A.1.3 sont ici testées.

- ***Classification de gestes 3D par MMCQ Pa***

Pour tester les performances de la méthode MMC Pa pour la classification de gestes 3D, une validation croisée "leave-one-out"a été exploitée [Hastie01]. Soit un ensemble de classes de gestes comprenant n gestes, les classes d'appartenances de ces gestes étant connues *a priori*. La classification d'un geste test s'effectue en utilisant les

$n-1$ trajectoires restantes pour l'apprentissage des classes de gestes. La même procédure est effectuée pour l'ensemble des n gestes. Le résultat de classification présenté correspond alors au pourcentage de gestes classé dans sa classe d'appartenance initiale.

Le taux de reconnaissance moyen, sur les deux ensembles de gestes utilisé, de la méthode de classification de gestes par MMCQ Pa par validation croisée "leave-one-out" est de 97.5%.

Ainsi, le résultat de classification obtenu montre l'efficacité de la méthode par MMCQ Pa lorsqu'il s'agit de comparer des gestes de tailles réduites.

• ***Clustering de gestes manuels 3D par MMCQ Pa***

Le clustering de gestes manuels a été abordé, avec notre méthode utilisant les MMCQ Pa. Les résultats présentés sont les résultats optimaux de clustering, *i.e.*, les résultats les plus intéressants parmi des partitions obtenues pour différentes valeurs du nombre de cluster désiré *a priori*. En appliquant une telle méthode, neuf clusters d'intérêt ont pu être exhibés, à la place des six clusters attendus, pour chacun des deux ensembles de 120 gestes.

En effet, trois des six classes de gestes correspondent à des gestes mono-manuels (*i.e.*, ici, les classes "pointer", "mouvement de négation" et "mouvement d'au revoir") alors que les trois autres classes (*i.e.*, les classes "nage", "vol" et "applaudissement"). Les classes composée de gestes mono-manuels étant composées d'acteurs gauchers et droitiers, le clustering opéré a divisé ces classes en deux clusters, un cluster pour les gauchers et un autre pour les droitiers effectuant ces gestes mono-manuels, les classes de gestes bi-manuels n'étant elles pas concernées par ce phénomène.

Ainsi, nous avons obtenu un clustering des deux ensembles de 120 gestes composé des neuf classes suivantes :

- mouvement de nage,
- mouvement de vol,
- applaudir,
- pointer du doigt, effectué par un droitier,
- mouvement de négation, effectué par un droitier,
- mouvement d'au revoir, effectué par un droitier,
- pointer du doigt, effectué par un gaucher,
- mouvement de négation, effectué par un gaucher,
- mouvement d'au revoir, effectué par un gaucher.

En considérant ces neuf classes, le résultat de clustering correct, en calculant la moyenne des résultats obtenu sur les deux ensembles de 120 gestes, atteint 89.2%. Ce résultat a été obtenu en fixant, pour la classification hiérarchique ascendante, le nombre de clusters désiré à 9. Ces résultats montre que les MMCQ Pa sont efficaces pour le clustering de gestes caractérisé par des trajectoires de tailles réduites.

A.1.4.3 Temps de calcul

Le calcul entre gestes à l'aide de la méthode MMCQ Pa est réduit. Par exemple, le temps moyen pour le calcul de la distance D_{gest} entre deux gestes 3D de 25 coordonnées chacun, avec un PC standard, est de l'ordre de 0.1 seconde. Ce temps de calcul comprend le calcul du nombre d'état des MMCQ Pa, de leurs paramètres ainsi que de leur comparaison.

A.1.5 Conclusion

Nous avons pu mettre en valeur la pertinence de la prise en compte des interactions entre parties du corps humain pour la reconnaissance de gestes. En effet, les interactions entre les mains, le torse et la tête, exprimées au travers de distances et d'angles, ont permis d'obtenir des résultats de classification et de clustering intéressants. Les gestes, correspondant à des mouvements de la partie supérieure du corps, sont de faibles durées (la taille moyenne de trajectoires associées à ces gestes est de 25 points). Nous ainsi pu mettre en valeur les propriétés des MMCQ Pa pour la modélisation des interactions entre trajectoires de tailles réduites.

A.2 Reconnaissance d'actions 3D par la prise en compte d'interactions entre membres du corps

Le but de cette section est de décrire une méthode de reconnaissance (supervisée ou non) d'actions. A la différence de la section précédente, les actions (et non plus les gestes) correspondent à des mouvements de l'ensemble du corps, et non pas seulement de la partie supérieure du corps humain. Ainsi, les actions sont portées par les mouvements de certaines parties du corps humain, chacun étant décrit par une trajectoire 3D correspondant aux positions successives dans l'espace. Les actions traitées pourront ainsi être des actions impliquant les membres inférieurs du corps humain, *i.e.*, des actions de marche, de saut, de course... De plus, les actions exploitées dans cette section étant de durée conséquente, les modélisations de type MMC/MMG pourront être exploitées, et plus particulièrement les MMC/MMG Pa (voir section 1.3.4).

Pour cette tâche de reconnaissance d'actions par analyse de trajectoires 3D, des données obtenues à l'aide d'outils de capture de mouvements ont été utilisées. La base de données correspondante peut être téléchargée [MoCapDATA] ; elle est associée à

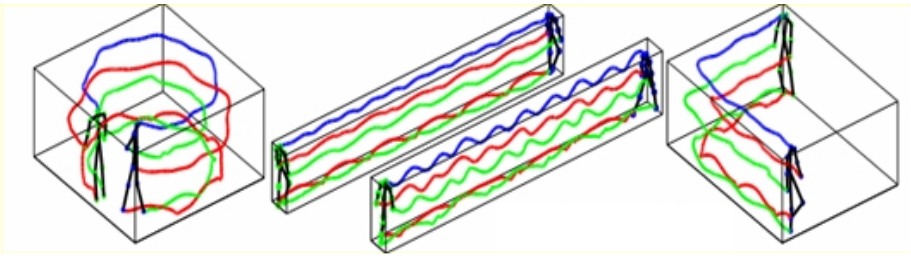

FIG. A.3 – Présentation des trajectoires obtenues par capture de mouvement, pour trois exemples de la classe d'action "marche", dans l'espace 4D (3D plus le temps).

l'article de Ali et al. [Ali 07]). La figure A.3, tirée de [Ali 07], présente un ensemble de trajectoires 3D associées aux différentes parties du corps. Le corps est décrit par 17 points, comme montré à la figure A.4. Ils correspondent aux genoux, aux coudes, à la tête... et ont été obtenus par un dispositif de capture de mouvements.

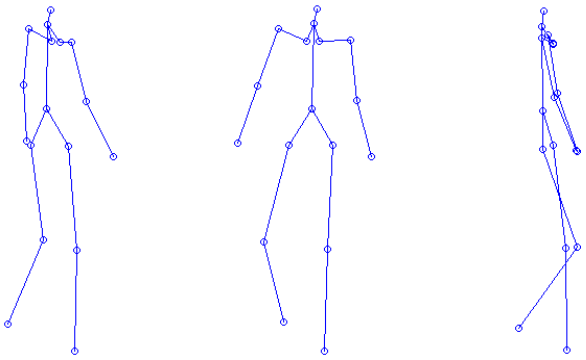

FIG. A.4 – Ensemble de 17 points représentant le corps dont les positions sont obtenues au cours du temps par un dispositif de capture de mouvement. Trois angles de vue sont donnés.

La première partie de cette section introduit la représentation des trajectoires 3D de parties du corps humain. La seconde partie sera dédiée à l'utilisation de MMC/MMG Pa pour la reconnaissance et le clustering d'actions.

A.2.1 Représentation d'actions 3D à l'aide d'interactions entre parties du corps humain

Nous avons opté pour une caractérisation des actions à l'aide des interactions entre parties du corps humain représentée par six variables :

- la distance entre les deux mains d_m,
- la distance entre les deux pieds d_p,

- les angles, dans le plan de profil, entre le corps et les jambes gauche et droite, notés θ_{JG} et θ_{JD} (voir figure A.5).

- les angles, dans le plan de profil, entre le corps et les bras gauche et droit, notés θ_{BG} et θ_{BD} (voir figure A.5).

Le plan de profil est le plan vertical au segment définit entre les deux épaules. Le corps a, lui, une orientation définie par le segment entre les deux points du ventre et du cou. Les jambes gauche et droite ont des directions données par les segments entre le ventre et les pieds correspondant. Enfin, les bras gauche et droit ont des directions données par le segment entre les épaules et les mains correspondantes. De plus, les formules des distances et des angles utilisés sont de la même forme que celle définies en section A.1.1.

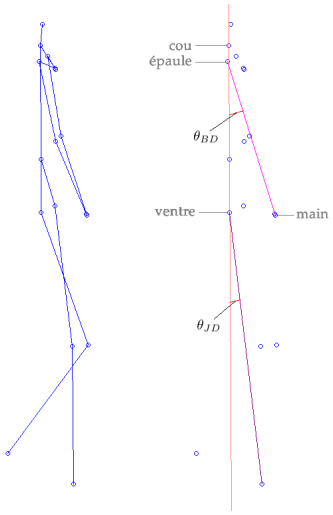

FIG. A.5 – Gauche : Vue des points définissant le corps humain, ainsi que leur liaisons, dans le plan de profil. Droite : illustration, dans le plan de profil, des angles θ_{JD} et θ_{BD}.

Les actions seront caractérisées par les valeurs centrées et normalisées des six variables définies, *i.e.*,

$$\tilde{d}_m = \frac{d_m - d_m^{mean}}{d_m^{mean}}, \tilde{d}_p = \frac{d_p - d_p^{mean}}{d_p^{mean}},$$

$$\tilde{\theta}_{JG} = \frac{\theta_{JG} - \theta_{JG}^{mean}}{\theta_{JG}^{mean}}, \tilde{\theta}_{JD} = \frac{\theta_{JD} - \theta_{JD}^{mean}}{\theta_{JD}^{mean}},$$

et

$$\tilde{\theta}_{BG} = \frac{\theta_{BG} - \theta_{BG}^{mean}}{\theta_{BG}^{mean}}, \tilde{\theta}_{BD} = \frac{\theta_{BD} - \theta_{BD}^{mean}}{\theta_{BD}^{mean}},$$

où les valeurs moyennes sont calculées pour chacune des actions.

Cette modélisation permet, d'assurer l'invariance aux possibles variabilités d'amplitude des actions, variabilité apparaissant du fait de la diversité possibles dans la réalisation d'une action ainsi que des différences physiologiques entre acteurs. De plus, cette représentation est invariante aux transformations, dans l'espace, de translations, de rotations (propriétés des distances relatives ainsi que des angles relatifs), sans besoin de recalage dans l'espace.

Les vecteurs caractérisant les actions sont les six vecteurs contenant les valeurs successives des six variables de représentation. Par exemple, le vecteur de caractérisation correspondant à la valeur $\tilde{d}_{JD}$ pour une action A_k (action d'index k composée de n_k images) sera :

$$V_{JD}^k = [\tilde{d}_{JD,1}^k, \tilde{d}_{JD,2}^k, ..., \tilde{d}_{JD,n_k-1}^k, \tilde{d}_{JD,n_k}^k].$$

A.2.2 Modélisation et comparaison d'actions 3D par MMC/MMG Parallèles

Les modélisations et comparaisons d'actions 3D sont modélisées à l'aide de MMC-/MMG Pa (voir section 1.3.4). En effet, contrairement aux gestes considérés en section précédente, les actions sont cette fois composées de plusieurs centaines d'observations. Cela permet d'estimer de façon fiable les paramètres des MMG pour la caractérisation d'actions. Ainsi, un MMC/MMG est défini pour chacune des six variables de représentation, et ce pour chacune des actions observées (de façon analogue à ce qui a été fait en section A.1.1). On pourra ensuite comparer cette approche aux MMCQ Pa (section A.1).

La méthode d'estimation *ICL* (voir section 2.3) est utilisée pour choisir les nombre d'états des composantes MMC/MMG du MMC/MMC Pa.

- ***Comparaison d'actions 3D par MMC/MMG Pa***

Afin de comparer deux actions, une mesure de similarité doit être définie entre les MMC/MMG Pa. Pour cela, nous avons recours à la distance croisée de Rabiner D_c entre MMC/MMG, définie en section 5.1.3. Nous faisons à nouveau l'hypothèse d'indépendance entre les six variables de caractérisation. La distance utilisée est ainsi la somme des distances entre les MMC/MMG respectifs composant le MMC/MMG Pa. La distance entre deux actions A_i et A_j est donc définie par :

$$D_{act}(A_i, A_j) = \sum_{l=1}^{6} D_s^l(\lambda_i^l, \lambda_j^l),$$

où l correspond à un index décrivant les six variables utilisées pour la représentation d'actions, et λ_i^l correspond aux paramètres (A, B, Π) du MMCQ associé à la variable d'index l pour l'action i.

- ***Reconnaissance supervisée et clustering d'actions 3D par MMC/MMG Pa***

Nous avons effectué la reconnaissance supervisée et non-supervisée d'actions 3D, à l'aide de trajectoires de parties du corps humain, modélisées par des MMC/MMG Pa de manière similaire aux reconnaissances supervisées et non-supervisées de gestes (section A.1.2), en exploitant cette fois la distance entre actions D_{act}.

A.2.3 Tests effectués et résultats obtenus

Nous allons appréhender maintenant une tâche de reconnaissance d'actions par analyse de trajectoires 3D. Pour cela, nous avons recours aux données issues de capture de mouvements disponibles en ligne [MoCapDATA]. Cette base de données a été introduite dans l'article de Ali et al. [Ali 07]). Nous pourrons ainsi comparer nos résultats aux leurs. La figure A.6, tirée de [Ali 07], présente les différentes classes que comprend la base de données. Les observations sont formées de séquences de coordonnées 3D, associées aux différentes parties du corps. Le corps est représenté par 17 points (figure A.4), placés aux genoux, aux coudes, à la tête... Ils sont obtenus par un dispositif de capture de mouvements. Les données disponibles contiennent cinq classes d'actions : "danse", "marche", "course", "saut" et "s'asseoir".

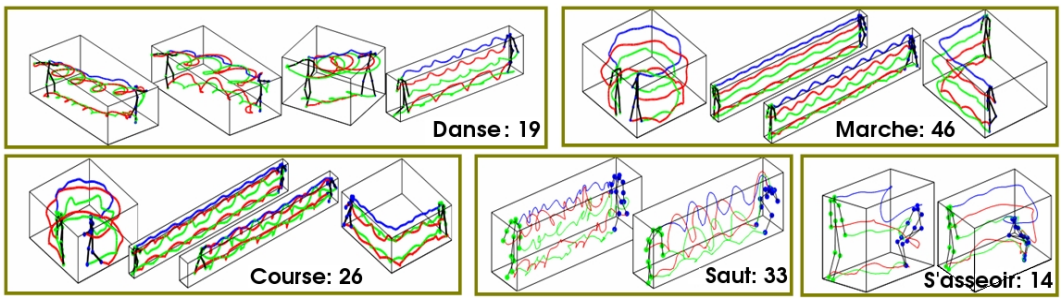

FIG. A.6 – Présentation d'exemples d'actions appartenant aux cinq classes, obtenus par capture de mouvement. Les trajectoires de cinq points sont tracées pour chaque action.

Tests non terminés.

Les résultats en cours de réalisation, sur la base de données de trajectoires issues de capture de mouvement (correspondant à la section A.2.3), permettront la comparaison

avec d'autres méthodes et notamment avec la méthode proposée par Ali et al. [Ali 07] pour la reconnaissance d'actions.

A.3 Extension de la méthode MCSM à la segmentation et la reconnaissance d'actions

Une extension naturelle et immédiate des modélisations proposées dans le chapitre 7 pour la segmentation de vidéos de sport consiste à modéliser des enchaînements, ou séquence, d'actions (par exemple, les actions 3D traitées en section A.2). Soit un ensemble de classes d'actions, chacune représentée par un ensemble de trajectoires. Soit également une séquence correspondant à un enchaînement d'actions 3D, chacune de ces actions appartenant aux classes d'actions connues. Le but est non plus seulement de reconnaître une action (tâche de reconnaissance, supervisée ou non), mais de reconnaître les actions ainsi que leur enchaînement. Peu de travaux se sont, jusqu'à aujourd'hui, penchés sur de tels problèmes combinant classification d'actions et reconnaissance de transitions entre actions [Morency 07].

Les caractérisations et modélisations des actions développées dans la section A.2 sont également utilisées, *i.e.*, les valeurs $\tilde{d}_m, \tilde{d}_p, \tilde{\theta}_{JG}, \tilde{\theta}_{JD}, \tilde{\theta}_{BG}$ et $\tilde{\theta}_{BD}$. La différence est que les (six) MMC/MMG Pa, pour chaque classe d'actions, sont appris à l'aide de l'ensemble des actions d'une même classe, contrairement aux MMC/MMG Pa utilisés en section A.2 qui appréhendaient une unique action.

De plus, les MCSM utilisés pour la segmentation de vidéos de sport sont ici réutilisés, les états S'_i étant décrits par les six MMG/MMC Pa associés à chacune des classes d'actions. Chacun des états S'_i correspond alors non plus à une phase d'activité de sport, mais à un classe d'action connues et apprises. La segmentation temporelle est alors obtenue, de manière similaire aux segmentations de vidéos de sport, à l'aide d'un algorithme de Viterbi pour MCSM.

Tests non terminés.

A.4 Conclusion

Les tâches de reconnaissance d'actions et de gestes 3D ont été traitées à l'aide de modélisations markoviennes parallèles, en considérant aussi bien des MMCQ Pa pour le traitement d'ensembles de données de faibles tailles que les MMC/MMG Pa. Des résultats intéressants et encourageants ont été obtenus pour la classification et le clustering de gestes 3D à l'aide des interactions entre parties du corps humain. Une extension de la méthode MCSM pour la segmentation et la reconnaissance d'actions a également été proposée.

De plus, des résultats, en cours de réalisation, de comparaison entre MMCQ et MMC/MMG pour la classification de gestes (ensembles de données de faibles tailles) et d'actions (ensembles de données de tailles conséquentes) devrait permettre, en plus des résultats déjà obtenus dans la partie II, de valider l'utilisation des MMCQ pour des ensembles de données de faibles tailles et des MMC/MMG pour des ensembles de grandes tailles.

Annexe B

Algorithmes de Viterbi, *forward-backward*, et de Baum-Welsh pour les modèles de Markov cachés

---

"Ayant rompu le lien filial qui nous rattachait à l'humanité, nous vivons. À l'estimation des hommes, nous vivons heureux ; il est vrai que nous avons su dépasser les puissances, insurmontables pour eux, de l'égoïsme, de la cruauté et de la colère ; nous vivons de toute façon une vie différente. [...] Aux humains de l'ancienne race, notre monde fait l'effet d'un paradis. Il nous arrive d'ailleurs parfois de nous qualifier nous-mêmes - sur un mode, il est vrai, légèrement humoristique - de ce nom de « dieux » qui les avait tant fait rêver. [...] l'ambition ultime de cet ouvrage est de saluer cette espèce infortunée et courageuse qui nous a créés. Cette espèce douloureuse et vile, à peine différente du singe, qui portait cependant en elle tant d'aspirations nobles. Cette espèce torturée, contradictoire, individualiste et querelleuse, d'un égoïsme illimité, parfois capable d'explosions de violence inouïes, mais qui ne cessa jamais pourtant de croire à la bonté et à l'amour. Cette espèce aussi, qui, pour la première fois de l'histoire du monde, sut envisager la possibilité de son propre dépassement ; et qui, quelques années plus tard, sut mettre ce dépassement en pratique. [...]
Ce livre est dédiée à l'homme."

Michel Houellebecq - ***Les particules élémentaires***

Nous traitons, dans ces annexes, de l'utilisation d'un MMC pour les problèmes suivants :

- la **reconnaissance d'une séquence** : étant donné une suite d'observations $y_{1:T}$ et un MMC, quelle est la séquence d'états $s_{1:T}$ sous-jacente la plus probable,

- la **probabilité d'observation d'une séquence** : étant donné une suite d'observations $y_{1:T}$ et un MMC, quelle est la probabilité que cet automate ait engendré la séquence d'observations $y_{1:T}$,

- l'**apprentissage** : étant donné une suite d'observations $y_{1:T}$, comment définir un MMC (au travers de ses paramètres) maximisant la probabilité d'observation de $y_{1:T}$ (*i.e.*, $p(y_{1:t}|\lambda)$).

Nous décrivons maintenant les algorithmes permettant de résoudre ces trois problématiques.

B.1 Reconnaissance d'une séquence et algorithme de Viterbi

Soit une séquence d'observations $y_{1:T} = (y_1, ..., y_T)$. Connaissant les paramètres d'un MMC, le but de l'algorithme de Viterbi est de trouver la séquence d'états $s_{1:T}$ sous-jacente la plus probable.

On considère, à l'instant t, la valeur $r_t(m)$ définie par :

$$r_t(m) = \max p(s_1, \ldots, s_{t-1}, \mathbf{s_t} = \mathbf{m}, y_1, \ldots, y_t).$$

L'algorithme de Viterbi calcule cette valeur sur l'ensemble des séquences $[s_1, \ldots, s_{t-1}]$ d'états possibles de la façon suivante :

- *Initialisation :*
  A l'instant $t = 1$,
  $$r_1(m) = \pi(m) b_m(y_1).$$
- *Récurrence :*
  Connaissant $r_{t-1}(m)$ pour les M états, on a :
  $$r_t(m') = \max_m r_{t-1}(m) a_{mm'} b_m(y_t).$$
  Ainsi, pour chacun des états m', on calcule $r_t(m')$ et on garde en mémoire un prédécesseur $q_t(m')$ défini par (Fig. B.1) :
  $$q_t(m') = \arg\max_m r_{t-1}(m) a_{mm'} b_{m'}(y_t).$$

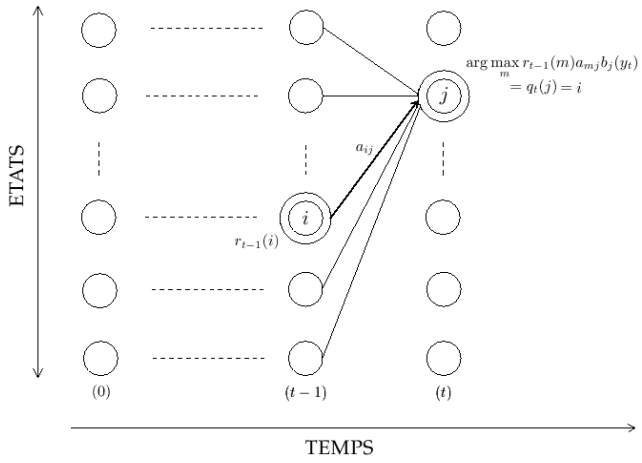

FIG. B.1 – Phase de récurrence de l'algorithme de Viterbi.

- *Finalisation :* Soit T la longueur de la séquence, $q_t(m)$ la fonction permettant d'obtenir le prédécesseur d'un état m à un instant t. L'état s_T à l'instant T est l'état m pour lequel $r_T(m)$ est le plus grand.

  La séquence d'états $s_{1:T}$ sous-jacente la plus probable est alors obtenue de façon récursive par rétropropagation (voir figure B.2) :

$$s_t = q_t(s_{t+1}), \forall t = T-1, \ldots, 1.$$

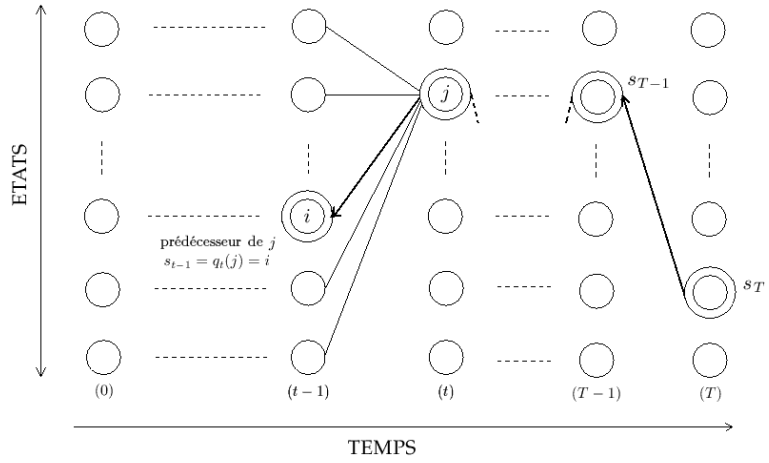

FIG. B.2 – Phase de finalisation de l'algorithme de Viterbi.

B.2 Probabilité d'observation d'une séquence et algorithme *forward-backward*

Le but de cette section est de présenter les méthodes de calcul de la probabilité d'observation d'une séquence associée à un MMC, *i.e.*, étant donné une suite d'observations $y_{1:T}$ et un MMC, quelle est la probabilité que ce modèle ait engendré la séquence d'observations $y_{1:T}$.

Un **premier algorithme** à considérer est le suivant :
soit $s_{1:T} = (s_1, ..., s_T)$ une suite d'états et $y_{1:T} = (y_1, ..., y_T)$ une suite d'observations. Avec l'hypothèse d'indépendance des observations, on a

$$\begin{aligned} p(y_{1:T}|s_{1:T}, \lambda) &= \prod_{t=1}^{T} p(y_t|s_t, \lambda) \\ &= b_{s_1}(y_1) b_{s_2}(y_2) \dots b_{s_T}(y_T). \end{aligned}$$

De plus, la probabilité associée à une séquence d'états donnée $s_{1:T} = (s_1, ..., s_T)$ est définie par

$$p(s_{1:T}|\lambda) = \pi_{s_1} a_{s_1 s_2} a_{s_2 s_3} \dots a_{s_{T-1} s_T}.$$

Enfin, on a

$$p(y_{1:T}, s_{1:T}|\lambda) = p(y_{1:T}|s_{1:T}, \lambda) p(s_{1:T}|\lambda). \tag{B.1}$$

Le calcul de la probabilité d'observation d'une séquence à un MMC se fait alors en énumérant les séquences d'états possibles $s_{1:T}$ et en sommant les probabilités décrites dans l'équation B.1 :

$$\begin{aligned} p(y_{1:T}|\lambda) &= \sum_{s_{1:T}} p(y_{1:T}, s_{1:T}|\lambda) \\ &= \sum_{s_{1:T}} p(y_{1:T}|s_{1:T}, \lambda) p(s_{1:T}|\lambda). \end{aligned}$$

Néanmoins, cette procédure s'avère trés coûteuse en termes de temps de calcul, et s'effectue en $O(TM^T)$ opérations ([Rabiner 89]), ce qui rend en pratique cet algorithme de calcul inutilisable.

Un **second algorithme** est alors à prendre en compte et s'appuie sur la valeur $\alpha_t(i)$, appelée variable *forward*, et définie par (Fig. B.3) :

$$\alpha_t(i) = p(y_{1:t}, s_t = i|\lambda).$$

La variable *forward* $\alpha_t(i)$ est la probabilité d'observer la séquence partielle $y_{1:t}$ pour l'état $s_t = i$, conditionnellement à λ.

L'algorithme de calcul des $\alpha_t(i)$ est le suivant :

- *Initialisation :*

$$\alpha_1(i) = \pi_i b_i(y_1).$$

- *Récurrence :*

$$\alpha_{t+1}(j) = \Big[\sum_{i=1}^{M} \alpha_t(i) a_{ij}\Big] b_j(y_{t+1})$$

- *Finalisation :*

$$p(y_{1:T}|\lambda) = \sum_{i=1}^{M} \alpha_T(i).$$

La complexité de cet algorithme, plus faible que celle du premier algorithme présenté, est de $O(M^2T)$ opérations ([Rabiner 89]), ce qui le rend utilisable, et ce quelle que soit la taille des séquences observées.

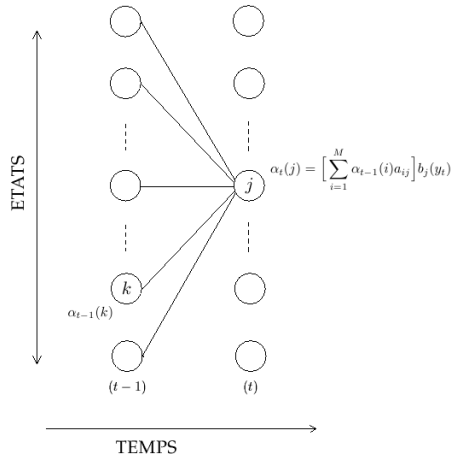

FIG. B.3 – Calcul de la variable *forward*.

Une seconde variable, appelée variable *backward* et notée $\beta_t(i)$, est définie par (Fig. B.4) :

$$\beta_t(i) = p(y_{t+1,T}|s_t = i, \lambda).$$

La variable *backward* $\beta_t(i)$ est la probabilité d'observer la séquence partielle $y_{t+1,T}$ sachant que l'état $s_t = i$. Nous décrivons ici cette variable car elle nous servira, en combinaison avec la variable *forward* $\alpha_t(i)$, dans la section suivante pour la présentation de l'algorithme de Baum-Welch pour l'apprentissage des paramètres d'un MMC.

L'algorithme de calcul des $\beta_t(i)$ est le suivant :

- *Initialisation :*

$$\beta_T(i) = 1$$

- *Récurrence :*

$$\beta_t(i) = \sum_{j=1}^{M} a_{ij} b_j(y_{t+1}) \beta_{t+1}(j), \quad 1 \leq i \leq M, \;\; t = T-1, \ldots, 1.$$

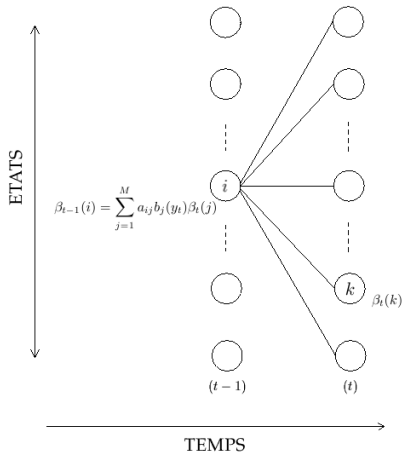

FIG. B.4 – Calcul de la variable *backward*.

B.3 Apprentissage et algorithme de Baum-Welch

Étant donné une suite d'observations $y_{1:T}$, comment définir un MMC (au travers de ses paramètres) maximisant la probabilité d'observation de $y_{1:T}$ (*i.e.*, $p(y_{1:T}|\lambda)$).

Il n'existe pas de solution analytique à ce problème complexe. Ainsi, nous présentons ici une méthode d'estimation itérative des paramètres d'un MMC, appelé algorithme de Baum-Welch [Baum 70, Dempster 77], et défini par :

1. Initialisation à λ_0 des paramètres du MMC.

2. Calculer un nouveau jeu de paramètres λ à partir de λ_0.

3. Si $log(p(y_{1:T}|\lambda)) - log(p(y_{1:T}|\lambda_0)) < \delta$, arrêt des itérations.

4. Sinon, le nouveau jeu de paramètres λ prend la place de λ_0 et on recommence à l'étape 2.

Afin d'effectuer cette procédure, il est nécessaire de définir le moyen de réestimer, à partir d'un jeu de paramètres λ_0, les paramètres λ du MMC.

Pour cela, on définit tout d'abord $\epsilon_t(i,j)$ comme la probabilité de se trouver dans l'état i au temps t et dans l'état j au temps $t+1$ conditionnellement à aux observations. Il apparaît immédiat, à partir des définitions des variables *forward* $\alpha_t(i)$ et *backward* $\beta_{t+1}(j)$, que $\epsilon_t(i,j)$ peut se calculer, comme l'illustre la figure B.5, par [Rabiner 89] :

$$\begin{aligned} \epsilon_t(i,j) &= \frac{\alpha_t(i)a_{ij}b_j(y_{t+1})\beta_{t+1}(j)}{p(y_{1:T}|\lambda)} \\ &= \frac{\alpha_t(i)a_{ij}b_j(y_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{M}\sum_{j=1}^{M}\alpha_t(i)a_{ij}b_j(y_{t+1})\beta_{t+1}(j)} \end{aligned}$$

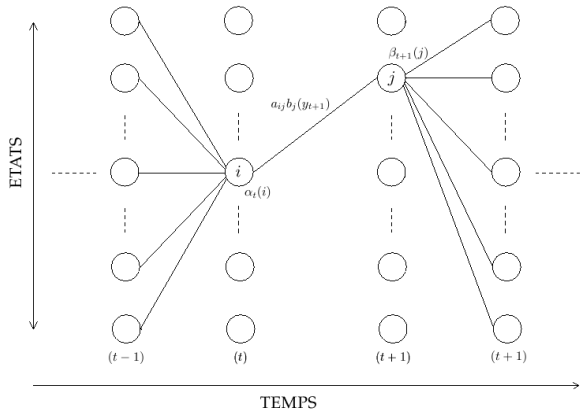

FIG. B.5 – Présentation de l'étape de calcul de ϵ dans l'algorithme de Baum-Welsh.

On utilise ensuite $\gamma_t(i)$, la probabilité d'être dans l'état i au temps t, donnée par

$$\gamma_t(i) = \sum_{j=1}^{M} \epsilon_t(i,j). \tag{B.2}$$

$\sum_{t=1}^{T} \gamma_t(i)$ correspond au nombre de fois où le processus est dans l'état i, et $\sum_{t=1}^{T-1} \epsilon_t(i,j)$ est le nombre de transitions de l'état i à l'état j au travers des T périodes.

On peut alors définir les règles de réestimation des paramètres $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ du MMC par :

- $\bar{\pi}_i = \gamma_1(i)$,

- $\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \epsilon_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$,

- $\bar{b}_j(k) = \bar{b}_{jk} = \frac{\sum_{t,s_t=k} \gamma_t(j)}{\sum_t \gamma_t(j)}$.

Il est à noter que les formules données ici sont valables pour la réévaluation des paramètres de B d'un MMC discret (à alphabet fini [Rabiner 89]) et donc B est dans ce cas une matrice. Pour des extensions de la phase de réestimation des paramètres à des MMC continus, le lecteur pourra se rapporter à [Rabiner 89, Bengio 99].

De plus, l'algorithme de Baum-Welsh assure la convergence vers un maximum local, mais non nécessairement vers le maximum global. Ainsi, les initialisations des paramètres d'un MMC peuvent être cruciales. Usuellement, les valeurs des paramètres de la matrice A et du vecteur π sont initialisées soit de façon uniforme, soit de façon aléatoire. Pour les valeurs à considérer dans B, le choix semble plus compliqué. Dans le cas discret, il apparaît comme important. Dans le cas continu, il est d'une importance critique et peut-être réalisé de manières diverses (en utilisant des méthodes de types *k-means* par exemple, algorithme présenté dans la section 2.4) [Rabiner 89].

"Bruno arriva vers vingt et une heures, il avait déjà un peu bu et souhaitait aborder des sujets théoriques. «J'ai toujours été frappé [...] par l'extraordinaire justesse des prédictions faites par Aldous Huxley dans "Le meilleur des mondes". Quand on pense que ce livre a été écrit en 1932, c'est hallucinant. Depuis, la société occidentale a constamment tenté de se rapprocher de ce modèle. Contrôle de plus en plus précis de la procréation, qui finira bien un jour par aboutir à sa dissociation totale d'avec le sexe, et à la reproduction de l'espèce humaine en laboratoire dans des conditions de sécurité et de fiabilité génétiques totales. Disparition par conséquent des rapports familiaux, de la notion de paternité et de filiation. Élimination, grâce aux progrès pharmaceutiques, de la distinction entre les âges de la vie. [...] Puis, quand il n'est plus possible de lutter contre le vieillissement, on disparaît par euthanaise librement consentie [...]. Il n'y a qu'une seule chose aujourd'hui qui heurte un peu notre système de valeurs égalitaire - ou plus précisément méritocratique - c'est la division de la société en castes, affectées à des travaux différents suivant leur nature génétique. Mais c'est justement le seul point sur lequel Huxley se soit montré mauvais prophète ; c'est justement le seul point qui, avec le développement de la robotisation et du machinisme, soit devenu à peu près inutile. [...] il a eu cette intuition - fondamentale - que l'évolution des sociétés humaines était depuis plusieurs siècles, et serait de plus en plus, exclusivement pilotée par l'évolution scientifique et technologique. [...] »

Bruno se resservit un verre de vin ; il commençait à avoir faim, et fut un peu surpris quand son frère lui répondit, d'une voix lasse : « [...] C'est une tradition anglaise, d'intellectuels pragmatiques, libéraux et sceptiques ; très différente du Siècle des lumières en France, beaucoup plus basée sur l'observation, sur la méthode expérimentale.»

Michel se leva, sortit de sa bibliothèque un volume intitulé "Ce que j'ose penser". « Il a été écrit par Julian Huxley, le frère ainé d'Aldous, et publié dès 1931, un an avant "Le meilleur des mondes". On y trouve suggérées toutes les idées sur le contrôle génétique et l'amélioration des espèces, y compris de l'espèce humaine, qui sont mises en pratique par son frère dans le roman. Tout cela y est présenté, sans ambiguïté, comme un but souhaitable, vers lequel il faut tendre. »"

Michel Houellebecq - ***Les particules élémentaires***

Table des figures

Bibliographie

[Agrawal 93] R. Agrawal, C. Faloutsos et A. Swami. Efficient similarity search in sequence databases. *Proc. of the Int. Conf. on Foundations of Data Organization and Algorithms*, Chicago, Etats-Unis, octobre 1993.

[Akaike 73] H. Akaike. Information theory as an extension of the maximum likelihood principle. *Second International Symposium on Information Theory*, Budapest, 1973.

[Aleotti 05] J. Aleotti et S. Caselli. Trajectory clustering and stochastic approximation for robot programming by demonstration. *Proc. of the IEEE Conf. on Intelligent Robots and Systems, IROS'05*, Edmonton, août 2005.

[Ali 07] S. Ali, A. Basharat et M. Shah. Chaotic invariant for human action recognition. *Proc. of the IEEE International Conference on Computer Vision, ICCV'05*, Pékin, octobre 2005.

[Alon 03] J. Alon, S. Sclaroff, G. Kollios et V. Pavlovic. Discovering clusters in motion time-series data. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition Conference, CVPR'03*, Toronro, juin 2003.

[Anjum 07a] N. Anjum et A. Cavallaro. Unsupervised fuzzy clustering for trajectory analysis. *Proc. of the IEEE Int. Conf. on Image processing, ICIP'07*, San Antonio, septembre 2007.

[Anjum 07b] N. Anjum et A. Cavallaro. Single camera calibration for trajectory-based behavior analysis. *Proc. of the IEEE Int. Conf. on Advanced Video and Signal based Surveillance, AVSS'07*, Londres, septembre 2007.

[Anjum 08] N. Anjum et A. Cavallaro. Multifeature object trajectory clustering for video analysis. *IEEE Trans. on Circuit and Systems for Video Technology*, Special issue on event analysis in videos, 18(11) :1555 :1564, novembre 2008.

[Antonini 06] G. Antonini et J.P. Thiran. Counting pedestrians in video sequences using trajectory clustering. *IEEE Trans. on Circuit and Systems for Video Technology*, 16(8) :1008-1020, août 2006.

[Assfalg 02] J. Assfalg, M. Bertini, A. Del Bimbo, W. Nunziati et P. Pala. Soccer highlights detection and recognition using hmms. *Proc. of the IEEE Int. Conf. on Multimedia and Expo, ICME'02*, Lausanne, août 2002.

[Audioseg] http ://gforge.inria.fr/projects/audioseg

[Bauer 06] D. Bauer, N. Brändle, S. Seer et R. Pfugfelder. Finding highly frequented paths in video sequences. *Proc. of the IEEE Int. Conf. on Pattern Recognition, ICPR'06*, Hong Kong, août 2006.

[Baum 70] L. E. Baum, T. Petrie, G. Soules et N. Weiss. A maximization technique occuring in the statistical analysis of probabilistic functions of Markov chains. *Annals of Math. Statistic.*, 41 :164-171, 1970.

[Barnard 05] M. Barnard et J-M. Odobez. Sports event recognition using layered HMMs. IDIAP-RR 05-07, 2005.

[Bashir 04] F. I. Bashir, A. A. Khokhar et D. Schonfeld. A hybrid system for affine-invariant trajectory retrieval. *Proc of the ACM SIGMM international workshop on Multimedia information retrieval table of contents*, New York, octobre 2004.

[Bashir 06] F. I. Bashir, A. A. Khokhar et D. Schonfeld. View invariant motion trajectory-based activity classification and recognition. *Multimedia Systems*, 12(1) :45-54, 2006.

[Bashir 07a] F. I. Bashir, A. A. Khokhar et D. Schonfeld. Real-time motion trajectory-based indexing and retrieval of video sequences. *IEEE Trans. on Multimedia*, 9(1) :58-65, 2007.

[Bashir 07b] F. I. Bashir, A. A. Khokhar et D. Schonfeld. Object trajectory-based activity classification and recognition using Hidden Markov Models. *IEEE Transactions on Image Processing*, 16(7) :1912-1919, 2007.

[bdforme] http ://www.imageprocessingplace.com/downloads_V3/root_downloads /image_databases/MPEG7_CE-Shape-1_Part_B.zip

[Bengio 95] Y. Bengio et P. Frasconi. An Input/Output HMM architecture. *Advances in Neural Information Processing Systems*, 7 :427-434, 1995.

[Bengio 99] Y. Bengio. Markovian models for sequential data. *Neural computing surveys*, 2 :129-162, 1999.

[Berclaz 08] J. Berclaz, F. Fleuret et P. Fua. Multi-camera tracking and atypical motion detection with behavorial maps. *Proc. of the European Conference on Computer Vision, ECCV'08*, Marseille, octobre 2008.

[Berndt 94] D. Berndt et J. Clifford. Using dynamic time warping to find patterns in time series. *Proc of KDD workshop*, Seattle, juillet 1994.

[Bernier01] O. Bernier et D. Collobert. Head and hands 3D tracking in real time by the EM algorithm. *Int workshop on recognition, analysis, and tracking of faces and gestures in real-time systems, RATFG-RTS'01*, Vancouver, juillet 2001.

[Biernacki 00] C. Biernacki, G. Celeux et G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(7) :719-725, juillet 2000.

[Bilmes 98] J. A. Bilmes. A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. International Computer Science Institute, Berkeley, avril 1998.

[Birgé 06] L. Birgé et Y. Rozenholc. How many bins should be put in a regular histogram. *ESAIM : Probability and statistics*, 10 :24-45, 2006.

[Black 98] M. J. Black et A. D. Jepson. A probabilistic framework for matching temporal trajectories : condensation-based recognition of gestures and expression. *Proc. of the European Conference on Computer Vision, ECCV'98* Fribourg, juin 1998.

[Bober 01] M. Bober. MPEG-7 visual shape descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6) :716-719, juin 2001.

[Bobick 96] A. F. Bobick et J. Davis. Real time recognition of activity using temporal templates. *IEEE Workshop on Applications of Computer Vision*, Sarasota, décembre 1996.

[Bobick 97] A. F. Bobick et A. D. Wilson. A state-based approach to the representation and recognition of gesture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12) :1325-1337, décembre 1997.

[Bobick 01] A. F. Bobick et J. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3) :257-267, mars 2001.

[Bouguet 99] J-Y. Bouguet. Pyramidal implementation of the lucas-kanade feature tracker description of the algorithm. Technical report, opencv documentation, Intel Corporation, Microprocessor Research Lab, 1999.

[Bourlard 97] H. Bourlard et S. Dupont. Subband-based speech recognition. *Proc. of the Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP'97*, Munich, avril 1997.

[Box 08] G. Box, G. M. Jenkins et G. C. Reinsel. Time series analysis : forecasting and control, 4th Edition. Wiley, Wiley Series in Probability and Statistics, 2008.

[Boykov 01] Y. Boykov et V. Kolmogorov. An experimental Comparison of min-cut/max-flow algorithms for energy minimization in computer vision. *Proc. of the Comp. Vision and Pattern Recognition Workshop on Energy Minimization Methods, EMMCVPR'01*, Sophia Antipolis, juin 2001.

[Brand 96] M. Brand et N. Oliver. Coupled hidden Markov models for complex action recognition. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'96*, San Francisco, Etats-Unis, pages 994-999, juin 1996.

[Brand 96] M. Brand. Coupled hidden Markov models for modeling interacting processes. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'96*, San Francisco, Etats-Unis, pages 994-999, juin 1996.

[Brand 00] M. Brand et V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) :844-851, août 2000.

[Bugeau 08] A. Bufeau et P. Pérez. Detection and segmentation of moving objects in highly dynamic scenes. *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'08*, Minneapolis, juin 2008.

[Buzan 04] D. Buzan, S. Sclaroff et G. Kollios. Extraction and clustering of motion trajectories in video. *Proc. of the IEEE Int. Conf. Pattern Recognition, ICPR'04*, pages 521-524, Cambridge, août 2004.

[Burges 98] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, Springer, 2 :121-167, 1998.

[Campbell 96] L. W. Campbell, D. A. Becker, A. Azarbayejani, A. F. Bobick et A. Pentland. Invariant features for 3-D gesture recognition. *Proceedings of the Conference on Automatic Face and Gesture Recognition, FG'96*, 157-163, Killington, octobre 1996.

[Chan 99] K. Chan et A. W. Fu. Efficient time series matching by wavelets. *Proc. of the IEEE Int. Conf. on Data Engineering*,Sydney, mars 1999.

[Chan 04] M. T. Chan, A. Hoogs, J. Schmiederer et M. Peterson. Detecting rare events in video using semantic primitives with HMM. *Proc. of the IEEE Int. Conf. on Pattern Recognition, ICPR'04*,pages 150-154, Cambridge, août 2004.

[Chan 06a] M. T. Chan, A. Hoogs, R. Bhotika et A. Perera. Joint recognition of complex events and track matching. *Proc of the IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'06*, pages 694-699, New York, juin 2006.

[Chan 06b] M. T. Chan, A. Hoogs, Z. Sun, J.Schmiederer, R. Bhotika et G. Doretto. Event recognition with fragmented object tracks. *Proc of the IEEE Int. Conf. on Pattern Recognition, ICPR'06*, Hong Kong, août 2006.

[Chang 98] S-F. Chang, W. Chen, H J. Meng, H. Sundaram et D. Zhong. A fully automated content-based video search engine supporting spatiotemporal queries. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5) :602-615, septembre 1998.

[Chang 02] P. Chang, M. Han et Y. Gong. Extract highlights from baseball game with hidden markov models. *Proc of the IEEE Int. Conf. on Image Processing, ICIP'02*, Rochester, septembre 2002.

[Charalampidis 05] D. Charalampidis. A modified k-means algorithm for circular invariant clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(12) :1856-1865, décembre 2005.

[Chen 00] W. Chen, S.-F. Chang. Motion trajectory matching of video objects. *Proc of the SPIE/IS&T Storage and Retrieval for Media Databases*, San Jose, janvier 2000.

[Chen 04] L. Chen, M. T. Özsu et V. Oria. Symbolic representation and retrieval of moving object trajectories. *Proc of the ACM SIGMM international workshop on Multimedia information retrieval table of contents*, New York, juin 2004.

[Chen 08] X. Chen, D. Schonfeld et A. Khokhar. Robust null representation and sampling for view-invariant motion trajectory analysis. *Proc of the IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'08*, Anchorage, juin 2008.

[Cheng 08] M. H. Cheng, M. F. Ho et C. L. Huang. Gait Analysis For Human Identification Through Manifold Learning and HMM. *Pattern Recognition*, 41(8) :2541-2553, août 2008.

[Comaniciu 02] D. Comaniciu et P. Meer. Mean shift : a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5) :603-619, Mai 2002.

[CVBASEDATA] http ://vision.fe.uni-lj.si/cvbase06/dataset.html

[CVBASEDOC] http ://vision.fe.uni-lj.si/cvbase06/downloads/CVBASE06manual .pdf

[Dailymotion] http ://www.dailymotion.com/fr

[Davis 93] J Davis et M. Shah. Recognizing hand gestures *Proceedings of the European Conference on Computer Vision, ECCV'94*, Stockholm, mai 1994.

[Dempster 77] A. P. Dempster, N. M. Laird et D. B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39 :1-38, 1977.

[Denman03] H. Denman, N. Rea, A. Kokaram. Content-based analysis for video from snooker broadcasts. *Computer Vision and Image Understanding*, 92(2-3) :176-195, décembre 2003.

[Ding 02] C. Ding, X. He, H. Zha et H. Simon. Adaptive dimension reduction for clustering high dimensional data. *Proc. of the IEEE Int. Conf. on Data Mining, ICDM'02*, Maebashi City, décembre 2002.

[Dockstader 03] S. L. Dockstader, N. S. Imennov et M. A. Tekalp. Markov-based failure prediction for human motion analysis. *Proc of the IEEE Int. Conf. on Computer Vision*, Nice, octobre 2003.

[Dockstader 06] S. L. Dockstader. Motion trajectory classification for visual surveillance and tracking. *Proc of the IEEE Int. Conf. on Advanced Video and Signal based Surveillance*, Sydney, novembre 2006.

[Durbin 98] R. Durbin, S. R. Eddy, A. Krogh, G. Mitchison. Biological Sequence Analysis : Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press. 1998.

[Durbin 01] J. Durbin et S.J. Koopman. Time series analysis by state space methods. Oxford Statistical Series, Oxford University Press, 2001.

[Faloutsos 94] C. Faloutsos, M. Ranganathan et Y. Manolopoulos. Fast subsequence matching in time-series databases. *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, Minneapolis, mai 1994.

[Fashandi 05] H. Fashandi et A.M.E. Moghaddam. A new rotation invariant similarity measure for trajectories. *Proc. of the IEEE Int. Symp. on Computational Intelligence in Robotics and Automation, CIRA'05*,pages 631-634, Espoo, juin 05.

[Feller 71] W. Feller. An Introduction to Probability Theory and Its Applications, Vol. 2, 3rd ed, Wiley, New York, 1971.

[Fine 98] S. Fine, Y. Singer et N. Tishby. The hierarchical hidden Markov model : analysis and applications. *Machine Learning*, 32 :41-62, 1998.

[Fourès 03b] T. Fourès et P. Joly. Defining Search Areas to Localize Limbs in Body Motion Analysis. *Int. Workshop on Adaptative Multimedia Retrieval, AMR'03*, Hambourg, septembre 2003.

[Fourès 03a] T. Fourès et P. Joly. A multi-level model for 2D human motion analysis and description. *International Symposium on Electronic Imaging Science and Technology 2003, EIST'03*, Santa Clara, janvier 2003.

[François 05] A. R. J. Francois, R. Nevatia, J. Hobbs, R. C. Bolles. VERL : An ontology framework for representing and annotating video events. *IEEE Multimedia Magazine*, pages 76-86, octobre-décembre 2005.

[Fraile 98] R. Fraile et S. J. Maybank. Vehicle trajectory approximation and classification. *Proc. of the British Machine Vision Conference*, Southampton, septembre 1998.

[Fritzke 95] B. Fritzke. A growing neural gas neural network learns topologies. *Advances in Neural Information Processing Systems*, MIT Press, 1995.

[Ford 98] J. Ford and J. Moore. Adaptive estimation of HMM transition probabilities. *IEEE Trans. on Signal Processing*, 46(5) :1374-1385, août 1998.

[Fu 01] T.-C. Fu, F.-L. Chung, V. Ng et R. Luk. Pattern discovery from stock time series using self-organizing maps. *Proc. of the ACM International Conference on Knowledge Discovery and Data Mining, SIGKDD'01*, San Francisco, août 2001.

[Fu 05] Z. Fu, W. Hu et T. Tan. Similarity based vehicle trajectory clustering and anomaly detection. *Proc. of the IEEE Int. Conf. on Image processing, ICIP'05*, Genève, septembre 2005.

[Fukunaga 75] K. Fukunaga et L. 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, janvier 1975.

[Ghahramani 97] Z. Ghahramani et M. Jordan. Factorial hidden Markov models. *Machine Learning* 29 :245-275, 1997.

[Gavrila 95] D. M. Gavrila et L. S. Davis. Towards 3-D model-based tracking and recognition of human movement : a multi-view approach. *Proc. of the Int. Work. on Automatic Face and Gesture Recognition, FG'95*, Zurich, juin 1995.

[Ge 02] X. Ge. Segmental semi-markov models and applications to sequence analysis. PhD thesis,. Université de Californie, Irvine, décembre 2002.

[Gengembre 08] N. Gengembre et P. Pérez. Probabilistic color-based multi-object tracking with application to team sports. Technical report , INRIA, RR-6555, mai 2008.

[Grabmeier 02] J. Grabmeier et A. Rudolph. Techniques of cluster algorithms in data mining. *Data Mining and Knowledge Discovery*, 6(4) :303-360,2002.

[Granger 69] C. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3) :424-438, 2003.

[Greig 89] D. Greig, B. Porteous et A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Stat. Soc.*, 51(2) :271-279, 1989.

[Grinias 02] I. Grinias et G.Tziritas. Robust pan, tilt and zoom estimation. *Proc. of the IEEE Int. Conf. on Digital Signal Processing, DSP'02*, Pine Mountain, octobre 2002.

[Gunsel 98] B. Günsel, A. M. Tekalp et P. J.L. van Beek. Temporal video segmentation using unsupervised clustering and semantic object tracking. *Journal of Electronic Imaging*, 7(3) :592-604, juillet 1998.

[Gunsel 99] B. Günsel, A. M. Tekalp et P. J.L. van Beek. Content-based access to video objects : temporal segmentation, visual summarization, and feature extraction. *Signal Processing*, Special Issue, 66(2) :261-280, avril 1998.

[Hakeem 07] A. Hakeem et M. Shah. Learning, detection and representation of multi-agent events in videos. Artificial Intelligence archive, 171(8-9) :586-605, juin 2007.

[Hamilton 94] J. Hamilton. Time series analysis. Princeton Univ Press, 1994.

[Han 01] J. Han et M. Kamber. Data Mining : Concepts and Techniques. Morgan Kaufmann, 2001.

[Härdle 04] W. Härdle, M. Muller, S. Sperlich et A. Werwatz. Nonparametric and semiparametric models. Springer, Springer series in statistics, Berlin, 2004.

[Hastie01] T. Hastie, R. Tibshirani et J. Friedman. The elements of statistical learning. Springer series in statistics, Berlin, 2001.

[Hervieu07a] A. Hervieu, P. Bouthemy et J-P. Le Cadre. A HMM-based method for recognizing dynamic video contents from trajectories. *Proc. of the IEEE Int. Conf. on Image Proc., ICIP'07*, San Antonio, septembre 2007.

[Hervieu07b] A. Hervieu, P. Bouthemy et J-P. Le Cadre. Reconnaissance d'événements dans des vidéos par l'analyse de trajectoires à l'aide de modèles de Markov. *Proc. du colloque GRETSI, GRETSI'07*, Troyes, septembre 2007.

[Hervieu 08a] A. Hervieu, P. Bouthemy et J-P. Le Cadre. Video event classification and detection using 2D trajectories. *Proc. of the Int. Conf. on Comp. Vis. Theory and Applications, VISAPP'08*, Madeira, janvier 2008.

[Hervieu 08b] A. Hervieu, P. Bouthemy et J-P. Le Cadre. Reconnaissance d'événements vidéos par l'analyse de trajectoires à l'aide de modèles de Markov. *Traitement du Signal*, sélectionné pour le numéro spécial GRETSI'07.

[Hervieu 08c] A. Hervieu, P. Bouthemy et J-P. Le Cadre. A statistical video content recognition method using invariant features on object trajectories. *IEEE Trans. on Circuit and Systems for Video Technology*, Special issue on event analysis in videos, 18(11), novembre 2008.

[Hervieu 08d] A. Hervieu, P. Bouthemy et J-P. Le Cadre. Activity-based temporal segmentation for videos of interacting objects using invariant trajectory features. *Proc. of the IEEE Int. Conf. on Image Proc., ICIP'08*, San Diego, octobre 2008.

[Hoey 00] J. Hoey et J. J. Little. Representation and recognition of complex human motion. *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition*, Hilton Head Island, juin 2000.

[Hongeng 00] S. Hongeng, F.Brémond et R. Nevatia. Representation and optimal recognition of human activities. *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition*, Hilton Head Island, juin 2000.

[Hongeng 01] S. Hongeng et R. Nevatia. Multi-agent event recognition. *Proc. of the Int. Conf. on Computer Vision, ICCV'01*, Vancouver, juillet 2001.

[Hongeng 03a] S. Hongeng, R. Nevatia et F. Bremond. Video-based event recognition : activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2) :129-162, 2003.

[Hongeng 03b] S. Hongeng, R. Nevatia et F. Bremond. Large-scale event detection using semi-hidden Markov models. *Proc. of the IEEE Int. Conf. on Computer Vision*, Nice, octobre 2003.

[Hu 04a] W. Hu, X. Xiao, T. Tan et S. Maybank. Learning activity patterns using fuzzy self-organizing neural network. *IEEE Transactions on Systems, Man and Cybernetics - Part B : Cybernetics*, 34(3) :1618-1626, juin 2004.

[Hu 04b] W. Hu, X. Xiao et T. Tan. . A hierarchical self-organizing approach for learning the patterns of motion trajectories. *IEEE Transactions on Neural Networks*, 15(1) :135-144, janvier 2004.

[Hu 06] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan et S. Maybank. A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9) :1450-1464, septembre 2006.

[Hu 07] W. Hu, D. Xie, Z. Fu, W. Zheng et S. Maybank. Semantic-based surveillance video retrieval. *IEEE Transactions on Image Processing*, 16(4) :1168-1181, avril 2007.

[Huber 81] P.J. Huber. Robust statistics. John Wiley and Sons, 1981.

[Hubert 74] L. Hubert. Approximate evaluation technique for the single-link and complete-link hierarchical clustering procedure. *Journal of the American Statistical Association*, 69(347) :698-704, 1974.

[Hung 07] H. Hung, D. Jayagopi, C. Yeo, G. Friedland, S. Ba, J-M. Odobez, N. Mirghafori, K. Ramchandran et D. Gatica-Perez. Using audio and video features to classify the most dominant person in a group meeting. *Proc. of the ACM Multimedia*, Augsburg, septembre 2007.

[Interactplay] http ://www.idiap.ch/resources/interactplay/index.php

[Ivanov 00] Y. A. Ivanov et A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 22(8) :852-872, août 2000.

[Izo 07b] T. Izo. Visual attention models for far-field scene analysis. These, Massashussets Institute of Technology, Cambridge, juin 2007.

[Izo 07a] T. Izo et W.E L. Grimson. Unsupervised modeling of object tracks for fast anomaly detection. *Proc. of the IEEE Int. Conf. on Image processing, ICIP'07*, San Antonio, septembre 2007.

[Jain 99] A. K. Jain, M. N. Murty et P. J. Flynn. Data clustering : a review. *ACM Comput. Surv.*, 31(3) :264-323, 1999.

[Johansson 73] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception and Psychophysics*, 73(2) :201-211, 1973.

[Johnson 95] N. Johnson et D. Hogg. Learning the distribution of object trajectories for event recognition. *Proc. of British Machine Vision Conf., BMVC'95*, Birmingham, juillet 1995.

[Joly 96] P. Joly et H.K.Kim. Efficient automatic analysis of camera work and microsegmentation of video using spatiotemporal images. *Signal Processing : Image Communication*, Elsevier, Eurasip, Amsterdam, 8(4), mai 1996.

[Joly 06] P. Joly. L'indexation des documents audiovisuels numériques. Habilitation à diriger des recherches, Université Paul Sabatier, décembre 2006.

[Junejo 04] I N. Junejo, O. Javed et H. Foroosh. Multi feature path modeling for video surveillance. *Proc of the IEEE Int. Conf. on Pattern Recognition, ICPR'04*, Cambridge, août 2004.

[Junejo 07a] I N. Junejo et H. Foroosh. Using Calibrated Camera for Euclidean Path Modeling. *Proc. of the IEEE Int. Conf. on Image processing, ICIP'07*, San Antonio, septembre 2007.

[Junejo 07a] I N. Junejo, E. Dexter, I. Laptev et P. Pérez. Cross-view action recognition from temporal self-similarities. *Proc. of the European Conference on Computer Vision*, Marseille, octobre 2008.

[Jung 08] C. R. Jung, L. Hennemann et S. R. Musse. Event detection using trajectory clustering and 4-D histograms. *IEEE Trans. on Circuit and Systems for Video Technology*, Special issue on event analysis in videos, 18(11) :1565 :1575, novembre 2008.

[Just 04] A. Just, O. Bernier et S. Marcel (2004). HMM and IOHMM for the recognition of mono- and bi-manual 3D hand gestures. *Proceedings of the British Machine Vision Conference, BMVC'04*, Kingston, septembre 2004.

[Kanungo 02] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman et A. Y. Wu. An efficient k-means clustering algorithm : analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :881-892, octobre 2002.

[Kahveci 01] T. Kahveci et A. Singh. Variable length queries for time series data. *Proc. of the IEEE Int. Conf. on Data Eng., ICDE'01*, Heidelberg, avril 2001.

[Keogh 00] E.J. Keogh et M.J. Pazzani. Scaling up dynamic time warping for data minig applications. In *Proc. of ACM int. conf. on Knowledge Discovery and Data Mining, SIGKDD'00*, Boston, août 2000.

[Keogh 01] E.J. Keogh et M.J. Pazzani. Derivative dynamic time warping. In *Proc. of SIAM Int. Conf. on Data Mining, SDM'01)*, Chicago, avril 2001.

[Keogh 02] E.J. Keogh. Exact indexing of dynamic time warping. In *Proc. of Int. Conf. on Very Large Databases,VLDB'02*, Hong Kong, août 2002.

[Khalid 05] S. Khalid et A. Naftel. Classifying spatiotemporal object trajectories using unsupervised learning of basis function coefficients. *ACM int. work. on Video Surveillance and Sensor Networks, VSSN'05*, Singapour, novembre 2005.

[Kijak 03] E. Kijak, L. Oisel et P. Gros. Hierarchical structure analysis of sports videos using hmms. *Proc of the IEEE Int. Conf. on Image Processing, ICIP'03*, Barcelona, septembre 2003.

[Kirshner 05] S. Kirshner. Modeling of multivariate time series using hidden Markov Models. Phd thesis, University of California, Irvine, 2005.

[Kohonen 97] T. Kohonen. Self-organizing maps. Springer-Verlag, Information Sciences Series, 1997.

[Kokaram 06] A. Kokaram, N. Rea, R. Dahyot, M. Tekalp, P. Bouthemy, P. Gros et I. Sezan. Browsing sports video (Trends in sports-related indexing and retrieval work). *IEEE Signal Processing Magazine*, 23(2) :47-58, mars 2006.

[Korn 97] F. Korn, H. Jagadish et C. Faloutsos. Efficiently supporting ad hoc queries in large datasets of time sequences. *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, Tucson, mai 1997.

[Lance 67] G. Lance et W. Williams. A general theory of classificatory sorting strategies. *Computer Journal*, 9 :373-380, 1967.

[Laptev05] I. Laptev. On space-time interest points. *IEEE International Journal of Computer Vision*, 64(2) :107-123, 2005.

[Laptev 07] I. Laptev et P. Pérez. Retrieving actions in movies. *Proc of the IEEE International Conference on Computer Vision, ICCV'07*, Rio de Janeiro, octobre 2007.

[Latecki 00] L. J. Latecki, R. Lakämper, et U. Eckhardt. Shape descriptors for non-rigid shapes with a single closed contour. *Proc of the IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'00*, Hilton Head Island, juin 2000.

[Lebarbier 04] E. Lebarbier et T. Mary-Huard. Le critère BIC : fondements théoriques et interprétation. *Rapport de recherche de l'INRIA*, RR-5315, septembre 2004.

[Lee 99] H.-K. Lee et J. H. Kim. An HMM-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10) :961-973, octobre 1999.

[Leonardi 02] R. Leonardi et P. Migliorati. Semantic indexing of multimedia documents. *IEEE Transactions on Multimedia*, 9(2) :44-51, avril 2002.

[Liao 05] T. W. Liao. Clustering of time series data - a survey. *Pattern Recognition*, 38 :1857-1874, 2005.

[Li 97] J. Li, A. Goralwalla, M. T. Özsu et D. Szafron. Modeling video temporal relationships in an object database management system. *Proc. of the IS&T/SPIE International Symposium on Electronic Imaging : Multimedia Computing and Networking*, San Jose, janvier 1997.

[Li 00] C. Li et G. Biswas. Bayesian clustering for temporal aata using hidden Markov model representation. *Proc. of the International Conference on Machine Learning, ICML'00*, Stanford, juillet 2000.

[Li 00b] J. Li, A. Najmi, R. M. Gray. Image classification by a two dimensional hidden Markov model. *IEEE Transactions on Signal Processing*, 48(2) :517 :533, février 2000.

[Li 02] C. Li et G. Biswas. Applying the hidden Markov model methodology for unsupervised learning of temporal data. *International Journal of Knowledge-based Intelligent Engineering Systems*, 6(3) :152-160, juillet 2002.

[Li 06] X. Li, W. Hu et W. Hu. A coarse-to-fine strategy for vehicle motion trajectory clustering. *Proceedings of the IEEE Int. Conf. on Pattern Recognition*, pages 591-594, hong-Kong, août 2006.

[Lin 04] J. Lin, M. Vlachos, E. Keogh et D. Gunopulos. Iterative incremental clustering of time series. *Proc. of the Conf. on Extending Database Technology, EDBT'04*, Crete, mars 2004.

[Liu 06] X. Liu et C.-S. Chua. Multi-agent activity recognition using observation decomposed hidden Markov models. *Image and Vision Computing*, 24 :166-175, février 2006.

[Lou 02] J. Lou, Q. Liu, T. Tan et W. Hu. Semantic interpretation of object activities in a surveillance system. *Proc. of the IEEE Int. Conf. on Pattern Recognition, ICPR'02*, pages 777-780, Quebec, août 2002.

[Makris 02] D. Makris et T. Ellis. Path detection in video surveillance. *Image and Vision Computing*, 20(12), pages 895-903, 2002.

[Makris 05] D. Makris et T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man and Cybernetics - Part B : Cybernetics*, 35(3) :397-408, juin 2005.

[Maliatski 05] B. Maliatski et Y.-P. Orly. Hardware-driven adaptive k-means clusterinf for real-time video imaging. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1) :164-166, janvier 2005.

[Mallat 99] S. Mallat. Towards ontology based cognitive vision. Academic Press, 1999.

[Marcel 00] S. Marcel, O. Bernier, J. E. Viallet et D. Collobert. Hand gesture recognition using Input-Output hidden Markov models. *Proceedings of the Conference on Automatic Face and Gesture Recognition, FG'00*, 456-461, Grenoble, mars 2000.

[McKenna 03] S. J. McKenna et K. Morrison. A comparison of skin history and trajectory-based representation schemes for the recognition of user-specified gestures. *Pattern Recognition*, 37 :999-1009, décembre 2003.

[Meek 02] C. Meek, D. M. Chickering et D. Heckerman. Autoregressive tree models for time-series analysis. *Proc of the SIAM Int. Conf. on Data Mining, SIAM ICDM'02*, Arlington, avril 2002.

[Melo 04] J. Melo, A. Naftel, A. Bernardino et J.S. Victor. Retrieval of vehicle trajectories and estimation of lane geometry using non-stationary traffic surveillance cameras. *Proc of the IEEE Int. Conf. on Advanced Concepts for Intelligent Vision Systems, ACIVS'04*, Bruxelles, août-septembre 2004.

[Mermelstein 76] P. Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern Recognition and Artificial Intelligence*, C. H. Chen, Ed., pp. 374-388, New York, 1976.

[Meyer 98] D. Meyer, J. Pösl et H. Niemann. Gait Classification with HMMS for Trajectories of Body Parts Extracted by Mixture Densities. *Proc of the British Machine Vision Conference, BMVC'98*, Southampton, septembre 1998.

[Minnen 03] D. Minnen et I. Essa et T. Starner. Expectation grammars : leveraging high-level expectations for activity recognition. *Proc of the IEEE Int. Conf. on Conputer Vision and Pattern Recognition, CVPR'03*, juin 2003.

[Mitiche 96] A. Mitiche et P. Bouthemy. Computation and analysis of image motion : a synopsis of current problems and methods. *International Journal of Computer Vision*, 19(1) :29-55, 1996.

[Min 04] J. Min et R. Kasturi. Activity recognition based on multiple trajectories. *Proc of the IEEE Int. Conf. on Pattern Recognition, ICPR'04*, Cambridge, août 2004.

[MoCapDATA] http ://www.cs.ucf.edu/ sali/Projects/ChaoticInvariants/index.html #Downloads

[Moënne-Loccoz 06] N. Moënne-Loccoz, E. Bruno et S. Marchand-Maillet. Local feature trajectories for efficient event-based indexing of video sequences. *Proc of the Int. Conf. on Image and Video Retrieval, CIVR'06*, Tempe, juillet 2006.

[Morency 07] L.-P. Morency, A. Quattoni et T. Darrell. Latent-dynamic discriminative models for continuous gesture recognition. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'07*, Minneapolis, juin 2007.

[Moore 01] D. Moore et I. Essa. Recognizing multiasked activities using stochastic context-free grammar. *Proc of the IEEE Conputer Vision and Pattern Recognition Workshop on Models vs Exemplars, CVPRWME'01*, Hawaii, décembre 2001.

[Murphy 02] K. P. Murphy. Dynamic bayesian networks : representation, inference and learning. These, Université de Californie, Berkeley, juillet 2002.

[Muscariello 09] A. Muscariello, G. Gravier et F. Bimbot. Variability tolerant audio motif discovery. *International Conference on Multimedia Modeling, MM'09*, Sophia-Antipolis, janvier 2009.

[Myers 81] C. S. Myers et L. R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected word recognition. *The Bell System Technical Journal*, 60(7) :1389-1409, septembre 1981.

[Nefian 02] A. Nefian, L. Liang, X. Pi, L. Xiaoxiang, C. Mao et K. Murphy. A coupled HMM for audio-visual speech recognition. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Orlando, mai 2002.

[Nevatia 03] R. Nevatia, T. Zhao et S. Hongeng. Hierarchical language-based representation of events in video streams. *IEEE Workshop on Event Mining*, Nice, octobre 2003.

[Nevatia 04] R. Nevatia, J. Hobbs et B. Bolles. An ontology for video event representation. *IEEE Workshop on Event Detection and Recognition, WEDR'04*, Washington, juillet 2004.

[Naftel 06a] A. Naftel et S. Khalid. Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. *Multimedia systems* 12(3) :227 :238, 2006.

[Naftel 06b] A. Naftel et F. B. Anwar. Visual recognition of manual tasks using object motion trajectories. *Proc. of the IEEE Int. Conf. on Video and Signal based Surveillance, AVSS'06*, Sydney, novembre 2006.

[Nascimento 08] J. C. Nascimento, M. A. T. Figueiredo et J. S. Marques. Unsupervised learning of motion patterns using generative models. *Proc. of the IEEE International Conference on Image Processing*, San Diego, octobre 2008.

[Natarajan 05] P. Natarajan et R. Nevatia. EDF : a framework for semantic annotation of video. *Proc. of the Workshop on Semantic Knowledge in Computer Vision*, Pékin, octobre 2005.

[Natarajan 07a] P. Natarajan et R. Nevatia. Hierarchical multi-channel hidden semi markov models. *Proc. of the IEEE Int. Joint Conference on Artificial Intelligence, IJCAI'07*, Hyderabad, janvier 2007.

[Natarajan 07b] P. Natarajan et R. Nevatia. Coupled hidden semi Markov models for activity recognition. *Proc. of the IEEE Workshop on Motion and Video Computing, WMVC'07*, Austin, février 2007.

[Nascimento 07] J. C. Nascimento, M. A. T. Figueiredo et J. S. Marques. Semi-supervised learning of switched dynamical models for classification of human activities in surveillance applications. *Proc. of the IEEE Int. Conf. on Image Processing, ICIP'07*, San Antonio, septembre 2007.

[Ng 02] A. Y. Ng, M. I. Jordan et Y. Weiss. On spectral clustering : analysis and an algorithm. *Advances in Neural Information Processing Systems, NIPS*, 14, 2002.

[Niu 04] W. Niu, J. Long, D. Han et Y. F. Wang. Human activity detection and recognition for video surveillance. *Proc of the IEEE Int. Conf. on Multimedia and Expo*, Taipei, juin 2004.

[Oates 01] T. Oates, L. Firoiu et P. R. Cohen. Using dynamic time warping to bootstrap HMM-based clustering of time series. *Lecture Notes In Computer Science*, 1828 :35-52, 2001.

[Odobez 94] J.-M. Odobez. Estimation, détection et segmentation du mouvement : une approche robuste et markovienne. Thèse de doctorat, Université de Rennes I, 1994.

[Odobez 95] J.-M. Odobez et P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4) :348-365, 1995.

[Okuma 04] K. Okuma, J. J. Little et D. G. Lowe. Automatic rectification of long image sequences. *Proc. of the Asian Conf. on Computer Vision, ACCV'04*, Ile de Jeju, janvier 2004.

[Oliver 00] N. M. Oliver, B. Rosario et A. P. Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 22(8) :831-843, août 2000.

[Pandit 83] M. S. Pandit et S. M. Wu. Time series and system analysis with applications. John Wiley & Sons, 1983.

[Pearl 88] J. Pearl. Probabilistc reasoning intelligent systems : networks of plausible inference. Morgan Kaufmann, 1988.

[Pérez 02] P. Pérez, C. Hue, J. Vermaak et M. Gangnet. Color-based probabilistic tracking. *Proc. Europ. Conf. Computer Vision, ECCV'02*, Copenhague, juin. 2002.

[Piciarelli 05] C. Piciarelli, G. L. Foresti et L. Snidaro. Trajectory clustering and its applications for video surveillance. *Proc. of the IEEE Int. Conf. on Advanced Video and Signal based Surveillance, AVSS'05*, pages 40-45, Côme, septembre 2005.

[Piciarelli 06] C. Piciarelli et G. L. Foresti. On line trajectory clustering for anomalous events detection. *Pattern Recognition Letters*, 27 :1835-1842, avril 2006.

[Piciarelli 07] C. Piciarelli et G. L. Foresti. Anomalous trajectory detection using support vector machines. *Proc. of the IEEE Advanced Video and Signal Based Surveillance, AVSS'07*, Londres, septembre 2007.

[Piciarelli 08] C. Piciarelli, C. Micheloni et G. L. Foresti. Trajectory-based anomalous event detection. *IEEE Trans. on Circuit and Systems for Video Technology*, Special issue on event analysis in videos, 18(11) :1544 :1554, novembre 2008.

[Piriou 06] G. Piriou and P. Bouthemy and J.-F. Yao. Recognition of dynamic video contents with global probabilistic models of visual motion. *IEEE Trans. on Image Processing*, 15(11) :3417 :3430, novembre 2008.

[Porikli 04a] F. Porikli. Trajectory distance metric using hidden Markov model based representation. *Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, Prague, mai 2004.

[Porikli 04b] F. Porikli. Trajectory pattern detection by HMM parameter space features and eigenvector clustering. *European Conference on Computer Vision, ECCV'04*, Prague, mai 2004.

[Porikli 04c] F. Porikli et T. Haga. Event detection by eigenvector decomposition using object and frame features. *Proc. of the Computer Vision and Pattern Recognition Workshop, CVPRW'04*, Washington, juin-juillet 2004.

[Popivanov 02] I. Popavinov et R. J. Miller. Similarity search over time series data using wavelets. *Proc. of the IEEE Int. Conf. on Data Engineering, ICDE'02*, San Jose, février-mars 2002.

[Povinelli 04] R. J. Povinelli, M. T. Johnson, A. C. Lindgren et J. Ye. Time series classification using Gaussian mixture models of reconstructed phase spaces. *IEEE Transactions on Knowledge and Data Engineering*, 16(6) :779-783, juin 2004.

[Prati 08] A. Prati, S. Calderara et R. Cucchiara. Using circular statistics for trajectory shape analysis. *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Anchorage, juin 2008.

[Psarrou 02] A. Psarrou, S. Gong et M. Walter. Recognition of human gestures and behaviour based on motion trajectories. *Image and Vision Computing*, 20 :349-358, février 2002.

[Rabiner 78] L. Rabiner et R. Schafer. Digital processing of speech signals. Prentice-Hall, Signal Processing Series, 1978.

[Rabiner 89] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77(2) :257-285, 1989.

[Rabiner 93] L. Rabiner and B. Juang. Fundamentals of speech recognition. Prentice Hall Signal Processing Series, 1993.

[Rao 02] C. Rao, A. Yilmaz et M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2) :203-226, novembre 2002.

[Raptis 08] M. Raptis, K. Wnuk et S. Soatto. Flexible dictionnaries for action classification. *Proc of the Workshop on Machine Learning for Vision-based Motion Analysis, MLVMA'08*, Marseille, octobre 2008.

[Remagnino 98] P. Remagnino, T. Tan et K. Baker. Agent orientated annotation in model based visual surveillance. *Proc. of the Int. Conf. on Computer Vision, ICCV'98*, Bombay, janvier 1998.

[Remagnino 01] P. Remagnino et G. A. Jones. Classifying surveillance events from attributes and behaviour. *Proc. of the British Machine Vision Conference, BMVC'01*, Manchester, septembre 2001.

[Richardson 06] M. Richardson et P. Domingos. Markov logic networks. *Machine Learning*, 62 :107-136, 2006.

[Robertson 05] N. Robertson et I. Reid. Behaviour understanding in video : a combined method. *Proc. of the IEEE Int. Conf. on Computer Vision, ICCV'05*, Pékin, octobre 2005.

[Schwarz 78] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics* 6 :461-464, 1978.

[Sebastian 01] T. B. Sebastian, P. N. Klein et B. B. Kimia. Recognition of shapes by editing shock graphs. *Proc. of the IEEE Int. Conf. on Computer Vision, ICCV'01*, Vancouver, Juillet 2001.

[Sebastian 04] T. B. Sebastian, P. N. Klein et B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5) :550-571, mai 2004.

[Siskind 96] J. M. Siskind et Q. Morris. A maximum-likelihood approach to visual event classification. *Proceedings of the European Conference on Computer Vision, ECCV'96*, 347-360, Royaume-Uni, avril 1996.

[Smith 99] J. R. Smith and S. F. Chang. Integrated spatial and feature image query. *Multimedia Systems*, 7(2) :129-140, mars 1999.

[Smyth 97] P. Smyth. Clustering sequences with hidden Markov models. *Advances in Neural Information Processing*, MIT Press, 1997.

[SPro] http ://gforge.inria.fr/projects/spro

[Starner 95] T. Starner. Visual recognition of American sign language using hidden Markov models. Master's Thesis, MIT, Février 1995.

[Starner 95] T. Starner et A. Pentland. Real-Time American Sign Language Recognition From Video Using Hidden Markov Models. *Technical Report* 375, MIT Media Lab, Perceptual Computing Group, 1995.

[Srivastava 03] A. Srivastava, S. Joshi, W. Mio et X. Liu. Statistical Shape Analysis : Clustering, Learning and Testing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4) :590-602, avril 2003.

[Srivastava 07] A. Srivastava, I. Jermin et S. Joshi. Riemannian Analysis of Probability Density Functions with Applications in Vision. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'07*, Minneapolis, juin 2007.

[Stardner 98] T. Stardner, J.Weaver et A. Pentland. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12) :1371-1375, décembre 1998.

[Stauffer 00] C. Stauffer et W. E. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) :747-757, août 2000.

[Shan 98] M. K. Shan et S. Y. Lee. Content-based video retrieval via motion trajectories. *Proc. of the SPIE Electronic Imaging and Multimedia System*, Bellingham, septembre 2004.

[Shi 94] J. Shi et C. Tomasi. Good features to track. *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'94*, Washington, juin 1994.

[Shi 00] J. Shi et J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Image Segmentation*, 22(8) :888-905, août 2000.

[Shi 04] Y. Shi, Y. Huang, D. Minnen, A. Bobick et I. Essa. Networks for recognition of partially ordered sequential action. *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'04*, Washington, juin 2004.

[Shim 04] C.-B. Shim, J.-W. Chang et Y.-C. Kim. Trajectory-based video retrieval for multimedia information systems. *International conference on advances in information systems*, Izmir, octobre 2004

[Sun 02] X. Sun, C.-W. Chen et B. S. Manjunath. Probabilistic motion parameter models for human activity recignition. *Proc of the Int. Conf. on Pattern Recognition, ICPR'02*, Québec, août 2002.

[Tran 08] S. D. Tran et L. S. Davis. Event modeling and recognition using Markov logic networks. *Proc. of the European Conference on Computer Vision, ECCV'08*, Marseille , octobre 2008.

[Van Wijk 99] J. J. Van Wijk, E. R. Van Selow. Cluster and calendar based visualization of time series data. *Proc. of the IEEE Symposium on Information Visualization, infovis'99*, San Francisco, octobre 1999.

[Vlachos 02] M. Vlachos, G. Kollios et D. Gunopulos. Discovering similar multidimensional trajectories. *Proc. of the IEEE Int. Conf. on Data Engineering, ICDE'02*, San Jose, février 2002.

[Vlachos 02b] M. Vlachos, D. Gunopulos et G. Kollios. Robust similarity measures for mobile object trajectories. *Proc. of the IEEE Int. Conf. on Database and Expert Systems Applications, DEXA'02*, Aix-en-Provence, septembre 2002.

[Vlachos 03] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos et E. Keogh. Indexing multidimensional time-series with support for multiple distance measures. *Proc. of the ACM Work. on Knowledge Discovery and Data Mining, SIGKDD'03*, Washington D.C., août 2003.

[Vlachos 03b] M. Vlachos, J. Lin, E. Keogh et D. Gunopulos. A wavelet-based anytime algorithm for k-means clustering of time series. *Proc. of the SIAM Int. Conf. on Data Mining*, San Francisco, mai 2003.

[Vlachos 06] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos et E. Keogh. Indexing multidimensional time-series. *Very Large Databases Journal*, 15(1) :1-20, 2006.

[Vogler 99] C. Vogler et D. Metaxas. Parallel hidden Markov models for American sign language recognition. *Proc. of the Int. Conf. on Computer Vision, ICCV'99*, Kerkyra, septembre 1999.

[Vogler01] C. Vogler et D. Metaxas. A framework for recognizing the simultaneous aspects of American sign language. *Computer Vision and Image Understanding*, 81(3) :358-384, mars 2001.

[Vu 02] V. -T. Vu, F. Bremond et M. Thonnat. Temporal Constraints for Video Interpretation. *Proc. of the European Conference on Artificial Intelligence, ECAI'02*, Lyon, juillet 2002.

[Wang 00] Y. Wang, Z. Liu et J. Huang. Multimedia content analysisusing both audio and visual clues. *IEEE Signal Processing Magazine*, 17(6) :12-36, 2000.

[Wang 06] X. Wang, K. Tieu et E. Grimson. Learning semantic scene models by trajectory analysis. *Proc. Europ. Conf. Computer Vision, ECCV'06*, Graz, mai 2006.

[Wang 08] X. Wang, K. T. Ma, G. W. Ng et E. Grimson. Trajectory analysis and semantic region modeling using a nonparametric Bayesian model. *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Anchorage, juin 2008.

[Waibel 89] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano et K. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3) :328-339, mars 1989.

[Weinland 07] D. Weinland, E. Boyer et R. Ronfard. Action recognition from arbitrary views using 3D exemplars. *Proc of the IEEE International Conference on Computer Vision, ICCV'07*, Rio de Janeiro, octobre 2007.

[Weiss 99] Y. Weiss. Segmentation using eigenvectors : a unifying view. *Proc. of the IEEE Int. Conf. on Computer Vision, ICCV'99*, Kerkira, septembre 1999.

[Wilpon 85] J. G. Wilpon et L. R. Rabiner. Modified k-means clustering algorithm for use in isolated word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 33(3) :587 :594, 1985.

[Wilson 99] A. D. Wilson et A. F. Bobick. Parametric hidden Markov models for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9) :884-900, 2002.

[Wu 00] Y. Wu, D. Agrawal et A. El Abbadi. A comparison of DFT and DWT based similarity search in time-series databases. *ACM Int. Conf. on Information and Knowledge Management, CIKM'00*, McLean, novembre 2000.

[Wren 97] C. R. Wren, A. Azarbayejani, T. Darrell et A. Pentland. Pfinder : real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7) :780 :785, juillet 1997.

[Xie 03] L. Xie, S.-F. Chang, A. Divakaran et H. Sun. Unsupervised discovery of multilevel statistical video structures using hierarchical hidden Markov models. *Proc. of the IEEE Int. Conf. on Multimedia and Expo, ICME'03* Baltimore ,juillet 2003.

[Xie 04] L. Xie, P. Xu, S.-F. Chang, A. Divakaran et H. Sun. Structure analysis of soccer video with domain knowledge and hidden Markov models. *Pattern Recognition Letters*, 25 :767 :775, février 2004.

[Yang 02] M. H. Yang, N. Ahuja et M. Tabb. Extraction of 2D motion trajectories and its application to hand gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8) :1061 :1074, août 2002.

[Yi 98] B. Yi, H. Jagadish et C. Faloutsos. Efficient retrieval of similar time sequences under time warping. *Proc of the IEEE Int. Conf. on Data Mining, ICDE'98*, Orlando, février 1998.

[Yilmaz05] A. Yilmaz et M. Shah. Recognizing human actions in videos acquired by uncalibrated moving cameras. *Proc of the IEEE International Conference on Computer Vision, ICCV'05*, Beijing, octobre 2005.

[Zelnicker 08] E. E.Zelnicker, S. Gong et T. Xiang. Global abnormal behaviour detection using a network of CCTV cameras. *Proc of the Workshop on Machine Learning for Vision-based Motion Analysis, MLVMA'08*, Marseille, octobre 2008.

[Zhang 04] D. Zhang, D. Gatica-Perez, S. Bengio et I. McCowan. Modeling individual and group actions in meetings with layered HMMs. *IEEE Trans. on Multimedia*, 8(3) :509-520, juin 2006.

[Zhang 07] Z. Zhang, K. Huang, T. Tan et L. Wang. Trajectory series analysis based event rule induction for visual surveillance. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'07*, Minneapolis, juin 2007.

[Zhang 08] Z. Zhang, K. Huang et T. Tan. Multi-thread parsing for recognizing complex events in videos. *Proc. of the European Conference on Computer Vision, ECCV'08*, Marseille, octobre 2008.

[Zhou 08] Y. Zhou, S. Yan et T. S. Huang. Pair-activity classification by bi-trajectories analysis. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'08*, Anchorage, juin 2008.

[Zhou 08b] Y. Zhou et T. S. Huang. 'Bag of segments' for motion trajectory analysis. *Proc. of the IEEE Int. Conf. on Image Processing, ICIP'08*, San Diego, octobre 2008.

Publications

Journaux internationaux

A. Hervieu, P. Bouthemy, et J-P. Le Cadre
A statistical video content recognition method using invariant features on object trajectories. - Dans *IEEE Trans. on Circuit and Systems for Video Technology*, Special issue on event analysis in videos, 18(11) :1533-1543, Novembre 2008.

Journaux nationaux

A. Hervieu, P. Bouthemy, et J-P. Le Cadre
Reconnaissance d'événements vidéos par l'analyse de trajectoires à l'aide de modèles de Markov. - dans *Traitement du signal*, à paraitre.

Congrès internationaux

A. Hervieu, P. Bouthemy, et J-P. Le Cadre
A HMM-based method for recognizing dynamic video contents from trajectories. Dans *Proc. of the IEEE Int. Conf. on Image Proc., ICIP'07*, San Antonio, Etats-Unis, Sept. 2007.

A. Hervieu, P. Bouthemy, et J-P. Le Cadre
Video event classification and detection using 2D trajectories. - Dans *Proc. of the Int. Conf. on Comp. Vis. Theory and Applications, VISAPP'08*, Madeira, Portugal, Jan. 2008.

A. Hervieu, P. Bouthemy, et J-P. Le Cadre
Activity-based temporal segmentation for videos of interacting objects using invariant trajectory features. *Proc. of the IEEE Int. Conf. on Image Proc., ICIP'08*, San Diego, Etats-Unis, Oct. 2008.

Congrès nationaux

A. Hervieu, P. Bouthemy, et J-P. Le Cadre
Reconnaissance d'événements dans des vidéos par l'analyse de trajectoires à l'aide de modèles de Markov. *Proc. du colloque GRETSI, GRETSI'07*, Troyes, France, Sept. 2007.

Résumé

Cette thèse s'intéresse à l'analyse du contenu dynamiques de vidéos à partir des trajectoires des objets mobiles extraites de ces vidéos.

L'approche développée est invariante à un certain nombre de transformations dans l'image, translation, rotation, facteur d'échelle, tout en prenant en compte simultanément des informations de dynamique et de forme sur les trajectoires. Un modèle de Markov caché (MMC) original est proposé qui permet notamment d'appréhender des situations où les observations sont en faible nombre (trajectoires courtes). La sélection automatique des paramètres est également abordée. Une mesure de similarité entre ces MMC a été définie et exploitée pour trois tâches de reconnaissance de contenus vidéo : la reconnaissance supervisée et le clustering de plans vidéo ainsi que la détection d'évènements rares. Des expérimentations ont été menées sur plusieurs ensembles de vidéos réelles de sport.

Des chaînes semi-markoviennes sont ensuite introduites afin de traiter les trajectoires de plusieurs objets en interaction. Les interactions entre trajectoires sont étudiées afin de reconnaître différentes phases d'activité. Notre méthode a été expérimentée avec succès sur des ensembles de trajectoires issues de vidéos de squash et de handball. De tels modèles ont été étendus à la reconnaissance de gestes et d'actions 3D, ainsi qu'à la segmentation temporelle d'actions. Les résultats montrent que la prise en compte des interactions pour de telles applications est d'un intérêt important.

Mots-clefs : trajectoires vidéos, reconnaissance d'évènements et de formes, détection d'évènements inattendus, reconnaissance d'activités, analyse de vidéos de sport.

Abstract

This thesis deals with the analysis of dynamic contents in videos. Our approach relies on trajectories extracted from the processed image sequences.

The developed method is invariant to translation, rotation and scaling while taking into account both shape and dynamics-related information on the trajectories. A novel hidden Markov model (HMM) framework is proposed which is able in particular to handle small sets of observations. Parameter setting is properly adressed. A similarity measure between the HMM is defined and exploited to tackle three dynamic video content understanding tasks : supervised recognition, clustering and detection of unexpected events. We have conducted experiments on several significative sets of real videos including sport videos.

Then, hierarchical semi-Markov chains are introduced to process trajectories of several interacting moving objects. The temporal interactions between trajectories are taken into account and exploited to characterize relevant phases of the activities in the processed videos. Our method has been favorably evaluated on sets of trajectories extracted from squash and handball videos. Applications of such interaction-based models have also been extended to 3D gesture and action recognition and clustering, and temporal segmentation of actions. The results show that taking into account the interactions is of great interest for such applications.

Keywords : video trajectories, recognition of events and shapes, detection of unexpected events, recognition of activities, sport video analysis.