Reconnaissance en-ligne d'actions 3D par l'analyse des trajectoires du squelette humain

Said Yacine Boulahia

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

INSA RENNES

THESE DE DOCTORAT DE

L'INSA RENNES

COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par

« Said Yacine BOULAHIA »

« Reconnaissance en-ligne d'actions 3D par l'analyse des trajectoires du squelette humain»

Thèse présentée et soutenue à « Rennes », le « 11.07.2018 »
Unité de recherche : IRISA - UMR6074

Rapporteurs avant soutenance :

Catherine ACHARD
Maître de conférences, Université Pierre et Marie Curie

Thierry PAQUET
Professeur des universités, Université de Rouen Normandie

Composition du Jury :

Catherine ACHARD
Maître de conférences, Université Pierre et Marie Curie/Rapporteur
Thierry PAQUET
Professeur des universités, Université de Rouen Normandie/Rapporteur
Indira THOUVENIN
Professeur des universités, Université de Technologie de Compiègne/Examinateur
Pierre-François MARTEAU
Professeur des universités, Université Bretagne Sud / Examinateur
Richard KULPA
Maître de conférences, Université Rennes 2 /Co-encadrant
Éric ANQUETIL
Professeur des universités, INSA de Rennes /Directeur de thèse

Remerciements

Je dédie cette modeste contribution à tous ceux, et j'ai de la chance car ils sont nombreux, m'ont permis de devenir ce que je suis. Ce mémoire serait trop volumineux pour les citer tous.

Pour commencer, j'adresse mes sincères remerciements aux rapporteurs, Catherine Achard et Thierry Paquet, et aux examinateurs, Indira Thouvenin et Pierre-François Marteau, qui ont accepté de faire partie du jury de thèse.

Je remercie tout naturellement mes encadrants : Éric Anquetil, Richard Kulpa et Franck Multon. Pendant les moments aussi bien de doute que d'enthousiasme, j'ai vécu ce que voulait dire faire équipe et j'en comprends maintenant la signification profonde : être complémentaire pour l'atteinte d'un objectif partagé. Encore une fois, merci.

Je remercie toute ma grande famille, pour les encouragements reçus aux moments les plus cruciaux de ma vie et, en ce moment, j'ai une pensée reconnaissante pour chacun d'eux.

Je remercie, bien sûr, mes amis de toujours, qui ont partagé avec moi, les bons moments mais aussi, les moins bons. Afin de n'oublier personne, je m'en tiendrai aux contrées : je pense donc bien à vous à Alger, Rennes, Oran, Paris, Sétif, Saint Malo, Tlemcen, Toulouse, Cancún (Mexique), Brno(Tchéquie), Washington D.C. (USA) ou Montréal (Canada). Je souhaiterais continuer à bénéficier, pendant longtemps encore, de leur compréhension, complicité et amitié.

Je remercie mes collègues de bureau, pour leur bonne humeur, et tous les membres des équipes Intuidoc et MimeTIC pour l'ambiance conviviale, notamment des pauses café.

Enfin, il m'est impossible de restituer tout ce que j'ai reçu et je fais le vœu d'apporter un fort soutien à ma famille, à mes amis sans oublier les inconnus que la vie mettrait sur mon chemin.

Table des matières

Chapitre 1

Introduction générale

Les évolutions des systèmes de capture des actions humaines et l'élargissement des domaines d'application où ces actions sont impliquées ont fait croître l'importance de l'interprétation automatisée des actions effectuées par l'humain. Ces applications s'étendent sur un large panel allant de la conception de nouvelles interfaces d'interaction jusqu'aux systèmes de vidéo surveillance, en passant par l'analyse du mouvement sportif, l'animation, les jeux vidéo, etc.

Ainsi, durant les dernières décennies, plusieurs approches pouvant reconnaître des actions 3D ont été proposées dans l'objectif d'améliorer l'interprétation des actions dans différents contextes. Ces approches ne traitent pas toutes de la même problématique et peuvent être répertoriées en trois familles. Nous retrouvons des approches qui se focalisent sur la modélisation des actions pré-segmentées, sur l'identification des actions dans un flot non segmenté ou la détection au plus tôt de ces actions (dite détection précoce).

Dans ce contexte, nous avons établi deux constats principaux. D'un côté, la plupart des travaux tentent de proposer des approches de plus en plus complexes en omettant souvent de tirer profit d'avancées réalisées dans d'autres domaines de la reconnaissance de formes. D'un autre côté, la plupart des travaux identifient souvent que très partiellement les difficultés sous-jacentes au problème adressé. En effet, il est peu fréquent de trouver une explication détaillée permettant de caractériser le problème en identifiant au préalable toutes les difficultés à relever pour en proposer ensuite des solutions appropriées. Autrement dit, la plupart des approches s'appuient sur une démarche de type "boîte noire" par opposition à l'approche que l'on a voulu défendre dans ces travaux qui peut être caractérisée comme une approche explicite revendiquant une certaine "transparence".

Au cours des travaux de cette thèse, nous avons abordé la problématique de recon-

naissance et de détection d'actions 3D en tentant d'élaborer une approche "transparente" en adressant explicitement chacune des difficultés identifiées. En particulier, nos travaux de recherche s'inscrivent dans une collaboration entre deux équipes de l'IRISA-Inria de Rennes, à savoir Intuidoc et MimeTIC. D'une part Intuidoc s'intéresse notamment à la conception de moteurs de reconnaissance de formes [AA11] et aux nouveaux usages autour de l'interaction gestuelle sur des surfaces tactiles. D'autre part, l'équipe de recherche MimeTIC travaille sur l'analyse, la modélisation et la simulation de la performance motrice humaine. MimeTIC associe des approches expérimentales pour mieux modéliser et simuler le mouvement humain corps-complet.

L'idée est de profiter de la complémentarité des savoir-faire des deux équipes de recherche en termes d'apprentissage, de modélisation et de représentation du geste, pour appréhender ce problème complexe sous un nouvel angle. Ainsi, nous proposons de reconsidérer les besoins et les difficultés rencontrées pour modéliser, reconnaître et détecter une action 3D en proposant de nouvelles solutions à la lumière des avancées réalisées en termes de modélisation de gestes manuscrits 2D au sein de l'équipe Intuidoc.

L'objectif final de nos travaux de recherche est de concevoir une approche "transparente" originale apte à détecter en temps réel l'occurrence d'une action, dans un flot non segmenté et idéalement le plus tôt possible. Pour parvenir à la mise en place d'une telle approche, nous avons adressé plusieurs sous-problèmes intermédiaires. Pour chacun de ces sous-problèmes, nous avons eu une démarche "transparente" et "explicative" dans les choix des stratégies retenues : (1) adresser méthodiquement les sous-problématiques permettant d'atteindre l'objectif final énoncé plus haut, (2) identifier explicitement les difficultés majeures à relever pour chaque sous-problématique, (3) proposer des solutions ciblant les difficultés relevées en se basant notamment sur le savoir-faire existant dans la communauté de la reconnaissance des formes 2D.

Pour atteindre notre objectif, nous avons fait émerger trois problématiques. La première consiste à réfléchir à une nouvelle approche pour modéliser et reconnaître **une action pré-segmentée**. En effet, il est d'abord nécessaire de développer une représentation à même de caractériser le plus finement possible une action donnée pour en faciliter la reconnaissance. Au cours de cette première étape, *dite reconnaissance pré-segmentée*, l'idée est de se focaliser sur la caractérisation d'une action en ayant connaissance du début et de la fin de cette dernière. Ainsi, la tâche de modélisation de l'action est décorrélée de celle de segmentation. La deuxième problématique consiste à développer une approche permettant de reconnaître **une action dans un flot non segmenté** : sans connaissance a priori du début ni de la fin de l'action. Cette deuxième problématique, *dite détection en-ligne*, est plus complexe que la première dans la mesure où il s'agit de reconnaître des actions tout en les segmentant dans un flot continu de mouvements, flot qui peut être

décomposé (ou non) par des positions de repos potentiellement identifiables. La troisième et dernière problématique consiste à concevoir une approche permettant la **caractérisation précoce d'une action avec très peu d'informations** (c'est-à-dire le début de l'action). Cette problématique vise à reconnaître au plus tôt une action en cours d'exécution, idéalement au début de cette action. Ainsi, cette tâche, *dite reconnaissance ou détection précoce*, permet de détecter ce qu'un utilisateur effectue comme action avant même qu'il l'ait achevée, ce qui est intéressant pour des systèmes d'interaction en temps réel. Mis bout à bout, les résultats issus de la résolution de chacune de ces problématiques permettent de répondre à l'objectif de ces travaux de thèse : à savoir la mise au point d'une approche "transparente" originale de reconnaissance et de détection d'actions dans un flot non segmenté, en s'inspirant des avancées réalisées en reconnaissance des formes 2D.

Pour chacune de ces trois problématiques, nous avons identifié explicitement les difficultés à considérer afin d'en effectuer une description complète pour permettre de concevoir des solutions ciblées pour chacune d'elles.

En particulier, pour la première problématique, **la reconnaissance d'actions pré-segmentées**, nous avons identifié trois difficultés majeures : la variabilité morphologique, les corrélations spatiales et le séquencement temporel. La **variabilité morphologique** représente la différence entre les données capturées pour une même classe d'action lorsqu'elle est effectuée par différents sujets. Ceci est expliqué notamment par la dépendance entre les données de capture de cette action et les caractéristiques anthropométriques du sujet l'ayant effectué (taille, envergure, etc.). **La corrélation spatiale** fait référence à la difficulté de modéliser les relations entre plusieurs articulations ou trajectoires (bras, jambes, etc.). En effet, lors de la performance d'une action 3D plusieurs articulations sont mises en jeu et la description et la modélisation des trajectoires associées servent à identifier l'action en cours. La difficulté est de concevoir des descripteurs qui permettent d'extraire suffisamment d'informations sur chacune de ces trajectoires tout en garantissant que la représentation finale de l'action ait une dimension réduite. La troisième et dernière difficulté est dite **séquencement temporel**. Elle fait référence à l'ordre dans lequel sont enchaînées les postures et représente en ce sens la corrélation cette fois temporelle entre les positions successives des articulations ou trajectoires. Nous avons explicitement adressé ces trois difficultés majeures dans la première partie de notre étude pour élaborer un moteur de reconnaissance d'actions pré-segmentées.

S'agissant de la deuxième problématique, celle de **la détection en-ligne d'actions dans un flot non segmenté**, nous avons identifié trois autres difficultés. En effet, en plus des trois difficultés de reconnaissance pré-segmentée, identifiées plus haut, il est nécessaire d'adresser la variabilité temporelle, la variabilité spatiale inter-classes et la

variabilité spatiale intra-classe. **La variabilité temporelle** traduit le fait qu'une même action peut être exécutée à des vitesses différentes et/ou avec des pauses insérées avant, pendant ou après l'action. Ceci est d'autant plus complexe que l'on ne connaît pas le début et la fin d'une action dans ce contexte de flot continu de mouvements. Cette variabilité empêche donc de connaître à l'avance la durée totale que prendrait une action qui serait en cours d'exécution. Ensuite, **la variabilité spatiale inter-classes** fait référence à la différence des déplacements articulaires qu'engendre la performance de chacune des classes d'actions (classe simple avec peu de mouvements - classe complexe avec beaucoup de mouvements). Ainsi, si cette variabilité n'est pas considérée, une décision (erronée) de détection peut être émise avant que l'action en cours ne soit réellement identifiable. Enfin, **la variabilité spatiale intra-classe** représente une propriété intrinsèque du mouvement humain. Elle conduit souvent au fait que des instances, appartenant pourtant à la même classe d'action, produisent des trajectoires de différentes amplitudes. On retrouve aussi cette difficulté dans la première problématique, mais elle est plus facile à appréhender étant donnée que les actions sont segmentées et que l'on peut alors opérer des procédures de normalisation. Ces trois variabilités sont identifiées comme des difficultés majeures que nous proposons de considérer lors de notre étude.

Pour **la détection précoce**, la troisième problématique de notre étude, une contrainte supplémentaire est à considérer, en plus de toutes celles évoquées plus haut. Il s'agit tout simplement de parvenir à détecter une action le plus tôt possible. Ainsi, en plus des difficultés relatives à une modélisation adéquate (reconnaissance) et celles relevant de la détection en flot non segmenté, il est nécessaire d'avoir une approche capable d'identifier et de caractériser une classe avec le moins d'informations possibles. Plus encore, il faut être capable de décider à partir de quand le système peut émettre une hypothèse de reconnaissance au plus tôt sans se tromper.

Nous considérons que l'identification et la formalisation de ces difficultés pour chacune des problématiques est déjà une contribution, dans la mesure où cela permet de délimiter les contours de ce défi scientifique. Plus encore, cette formalisation pourrait servir de cahier des charges pour des travaux futurs.

Après l'identification des principales difficultés à relever pour chacune des trois problématiques, il devient possible de concevoir des approches "transparentes" adressant explicitement ces problématiques. Comme évoqué au début de cette introduction, nos approches ont la particularité de tirer profit du savoir-faire de la reconnaissance des formes 2D, notamment les travaux portant sur la reconnaissance d'écriture et de tracés manuscrits 2D.

Ainsi, pour résoudre la première problématique portant sur la reconnaissance d'actions pré-segmentées, nous proposons dans un premier temps de concevoir une représentation

permettant de modéliser une action 3D pré-segmentée à base de données squelettiques. Dans ce travail, nous avons exploré deux pistes pour mener le transfert du savoir-faire 2D vers la 3D, chacune résultant d'une représentation différente des trajectoires de mouvements déduites des données squelettiques.

Dans un second temps, pour répondre à la problématique de détection en-ligne d'actions squelettiques dans un flot non segmenté, nous introduisons le concept de **segmentation curviligne des trajectoires** en opposition au concept existant de fenêtres temporelles. Ce nouveau concept de segmentation a la particularité de considérer les frames en entrée non pas comme un flot temporel mais comme un flot spatial. Ce concept permet notamment de faire face à la variabilité temporelle, première difficulté relevée pour cette problématique. Par ailleurs, nous proposons de concevoir un système de décision innovant, combinant les détections locales de plusieurs classifieurs spécialisés lancés en parallèle. Nous expliquons comment ce système permet d'adresser les deux difficultés restantes, à savoir la variabilité spatiale inter-classes et la variabilité spatiale intra-classe.

Enfin, nous proposons d'étendre l'approche de détection en-ligne pour adresser la dernière problématique : la détection précoce d'actions dans un flot non segmenté. Pour ce faire, nous avons mis en œuvre une approche constituée de trois modèles, chacun se basant sur le concept de segmentation curviligne. Ces modèles vont traiter le flot d'entrée à court, moyen et long terme de façon à réduire à la fois le risque d'erreur mais aussi la latence.

Les représentations et les modèles conçus feront l'objet d'évaluations expérimentales variées sur plusieurs benchmarks de l'état de l'art, de manière à mettre en exergue leurs propriétés. En particulier, nous rapporterons des résultats obtenus sur six benchmarks en effectuant des comparaisons avec une vingtaine d'approches de l'état de l'art.

Les chapitres structurant ce manuscrit de thèse s'appuient sur les différentes problématiques décrites précédemment :

Chapitre 2 : De nombreuses études ont été menées dans le cadre de la modélisation, la reconnaissance et la détection d'actions 3D. Nous proposons dans ce chapitre une synthèse de ces travaux en nous focalisant sur les approches utilisant des données squelettiques. Après une présentation en trois temps de ces approches, nous proposons une discussion détaillant les constats tirés de cette étude bibliographique, quelles sont les insuffisances relevées et les solutions que nous prévoyons de mettre en œuvre.

Chapitre 3 : Ce chapitre est focalisé sur la première problématique considérée, à savoir **la modélisation et la reconnaissance d'actions 3D pré-segmentées** à base de données squelettiques. La reconnaissance d'actions 3D est en effet un des sujets de recherche les plus actifs dans le domaine de la vision par ordinateur et la reconnaissance de formes. Nous présentons dans ce chapitre les difficultés de cette tâche, et proposons par

la suite deux nouvelles représentations d'actions 3D pré-segmentées inspirées de la reconnaissance de tracés et symboles 2D. Nous présentons également les résultats obtenus sur trois benchmarks d'actions 3D pré-segmentées squelettiques suivant plusieurs protocoles d'évaluation.

Chapitre 4 : Nous nous sommes intéressés dans ce chapitre à la **détection d'actions 3D**. Nous proposons en particulier une nouvelle approche de segmentation en-ligne à base de distance curviligne en opposition aux approches à base de fenêtres temporelles classiques. Cette approche est ensuite étendue pour adresser le problème de la détection au plus tôt dite **détection précoce** au moyen de trois modèles curvilignes. L'évaluation expérimentale est conduite sur quatre benchmarks d'actions non segmentées suivant différents protocoles d'évaluation, afin de se comparer aux approches de l'état de l'art dans le contexte de la détection en-ligne et celui de la détection précoce.

Chapitre 5 : Ce chapitre est consacré à la présentation de trois applications (showcase) dans lesquelles nous avons mis en œuvre les approches de reconnaissance et de détection proposées dans les chapitres précédents. Tout d'abord, nous présentons une nouvelle approche de reconnaissance de gestes dynamiques de la main. Dans un deuxième temps, nous présentons un système d'interaction sous forme d'un jeu. Dans une dernière section, nous analysons les résultats d'une étude évaluant nos approches pour améliorer l'animation d'humain virtuel en temps réel.

Chapitre 6 : Le dernier chapitre consiste en une conclusion globale de ce manuscrit, résumant l'ensemble des problématiques abordées au cours des travaux de cette thèse ainsi que les principales solutions proposées. Nous présentons également dans ce chapitre les perspectives qui se sont dégagées au cours des études menées. Il s'agit en fait d'améliorations permettant l'élaboration d'approches et de systèmes encore plus performants, notamment en termes de capacité de modélisation et de réduction de la latence de détection.

Chapitre 2

État de l'art

Ce chapitre est consacré à l'étude de la bibliographie portant sur la modélisation et la reconnaissance d'actions 3D dans un flot segmenté ou non. Cette étude est présentée en cinq sections.

La première section offre une vue d'ensemble des quatre aspects suivant lesquels notre étude des travaux de l'état de l'art est structurée. Chacun de ces aspects porte sur une étape du processus de reconnaissance d'actions, de sorte à analyser les approches précédentes de bout en bout. Ces sections comportent aussi des discussions sur les avantages et les limites des approches et plus généralement sur le contexte de leur utilisation.

La deuxième section porte sur la typologie des données reçues en entrée par les systèmes de reconnaissance d'actions. De par le contexte de nos travaux de recherche, nous focalisons cette étude sur les différentes natures de données squelettiques pouvant être utilisées.

La troisième section comporte une description des différentes techniques de modélisation et de classification des actions. Nous distinguons principalement les approches séquentielles, tenant compte implicitement de la nature séquentielle d'une action, des approches statistiques, qui intègrent explicitement cet aspect dans la représentation produite.

La quatrième section est consacrée à la dernière étape du processus, à savoir la reconnaissance d'actions dans un flot non segmenté dite *détection*. Il s'agit notamment d'expliciter les procédures de segmentation en temps réel utilisées par les approches précédentes.

Enfin, nous rapportons dans la conclusion de ce chapitre plusieurs constations relevées au cours de cette étude et nous délimitons ensuite le cadre global dans lequel s'inscrivent nos travaux de thèse.

2.1 Vue d'ensemble

Comme évoqué en introduction, l'objectif de cette thèse est de concevoir une approche "transparente" originale apte à reconnaître en temps réel l'occurrence d'une action, dans un flot non segmenté et idéalement le plus tôt possible. Une telle approche est notamment utile pour développer des systèmes d'interaction homme-machine, de plus en plus demandés ces dernières années.

Avant d'entreprendre la vue d'ensemble des approches de reconnaissance, il convient d'abord de préciser ce qu'est une action. En réalité, le sens attribué au mot action (ou geste) varie en fonction du domaine d'utilisation concerné. Plusieurs définitions ont donc été proposées dont la distinction principale réside dans leur niveau d'abstraction sémantique.

Dans ce manuscrit, nous retenons en particulier la distinction proposée par [Ram91, NSMG03, CW00] et qui définit l'action selon deux approches, **fonctionnelle** et **phénoménologique**. D'une part, l'approche fonctionnelle se réfère aux fonctions qu'une action peut exécuter dans des situations spécifiques, ce qui correspond à la **sémantique** de l'action. D'autre part, l'approche phénoménologique est fondée sur des critères cinématiques, spatiaux et fréquentiels qui décrivent le mouvement associé à cette action.

L'objectif d'une approche de reconnaissance d'actions est de faire correspondre la cinématique d'une action à sa sémantique. En effet, une approche de reconnaissance permet de combler le fossé dit **sémantique** qui représente la différence de niveau conceptuel entre la machine, qui opère sur des données cinématiques (données brutes), et l'utilisateur qui comprend la signification d'une action. Un telle approche propose donc une interprétation haut-niveau (la classe de l'action, l'intention de l'action) correspondant aux données bas-niveau issues de systèmes de capture.

De tels systèmes de reconnaissance ont un rôle crucial dans plusieurs domaines d'application centrés sur l'humain, notamment l'analyse vidéo [GCR12], la surveillance [JCK13], la robotique [DKOH15], l'interaction homme-machine [SDD12], la réalité augmentée et virtuelle [GBCC08], l'assistance aux personnes âgées [OOH$^+$05], les maisons intelligentes [BLMC09], l'éducation [MBR$^+$09], les jeux sérieux [Fuj00], etc.

Un système de reconnaissance opère plusieurs traitements sur les données reçues en entrée en phase d'apprentissage et en phase de test comme illustré dans la Figure 2.1. Les données relatives à l'action effectuée sont d'abord collectées via un système de capture. Ces données sont ensuite pré-traitées notamment pour éliminer le bruit et réduire la dépendance de ces données aux systèmes de capture utilisés et aux caractéristiques anthropométriques du sujet ayant effectué l'action. De ces données pré-traitées, des représentations sont extraites pour modéliser l'action. L'objectif de la construction de telles représenta-

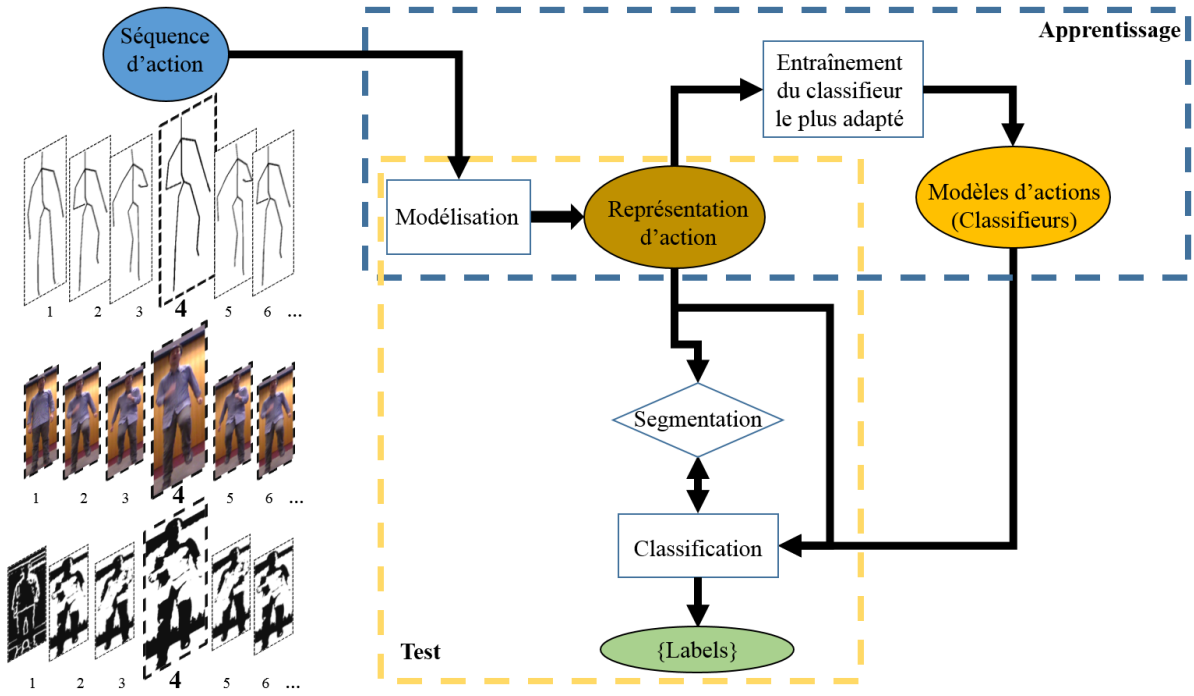

FIGURE 2.1 – Diagramme de flux de données pour un système générique de reconnaissance d'actions, comprenant des étapes interdépendantes de modélisation, d'entraînement, de segmentation et de classification.

tions est d'extraire des informations descriptives compactes pour coder et caractériser les attributs d'une action à partir de données de capture (par exemple forme humaine, pose et mouvement). Enfin un classifieur est entraîné avec ces représentations pour apprendre les dépendances entre elles et les classes d'actions auxquelles elles appartiennent. En phase de test, le moteur de reconnaissance ainsi appris est supposé reconnaître l'appartenance de nouveaux échantillons non utilisés pendant la phase d'apprentissage.

En analysant la Figure 2.1 de gauche à droite, il est d'abord possible de noter qu'une telle approche peut se baser sur des données de différentes natures. En effet, les approches de reconnaissance d'actions peuvent avoir en entrée une séquence d'images RGB, une séquence de cartes de profondeurs ou une séquence de squelette humain.

Dans le cadre de cette thèse nous utilisons uniquement des séquences de squelette humain et cela pour diverses raisons. Le concept de représentation basée sur le squelette remonte aux travaux de Johansson [Joh73], qui a démontré que la perception des mouvements d'un nombre réduit d'articulations squelettiques permet d'identifier les actions 3D d'un humain. Les représentations 3D basées sur des squelettes ont permis également d'atteindre des performances prometteuses dans diverses applications, dont les jeux basés sur Kinect, ainsi que dans le domaine de vision par ordinateur [DWW15, IBB]. De plus,

les représentations 3D basées sur le squelette sont capables de modéliser la relation entre les articulations humaines et d'encoder la configuration du corps entier. Elles sont également robustes aux changements d'échelle et d'illumination, et peuvent être invariantes à la position de caméra mais aussi à la rotation de corps humain. En outre, de nombreuses représentations basées sur des squelettes peuvent être construites à une fréquence élevée, ce qui peut grandement faciliter les applications temps réel. De ces faits, nous nous limitons dans notre étude bibliographique aux approches utilisant des séquences de squelette humain.

Les données squelettiques peuvent néanmoins être de différentes natures. Comme illustré dans la Figure 2.2, il peut s'agir de positions 3D des articulations constituant le squelette ou bien de leurs angles. De plus, pour modéliser une action, ces données peuvent être utilisées soit en tant que coordonnées absolues soit pour calculer des données dites relatives qui reflètent uniquement la progression ou le changement des coordonnées brutes les unes par rapport aux autres. Enfin, certaines représentations proposent de se baser sur les relations géométriques (sous forme booléenne) entre les parties du corps (voir Figure 2.12). De ce fait, un premier critère de différentiation des approches de reconnaissances est le type de données utilisées en entrée : les coordonnées cartésiennes (absolues ou relatives), les coordonnées angulaires (absolues ou relatives), les relations géométriques ou une combinaison de ces modalités.

Le deuxième critère de différentiation entre les approches de reconnaissance est relatif à la manière d'appréhender la nature séquentielle des données gestuelles. En effet,

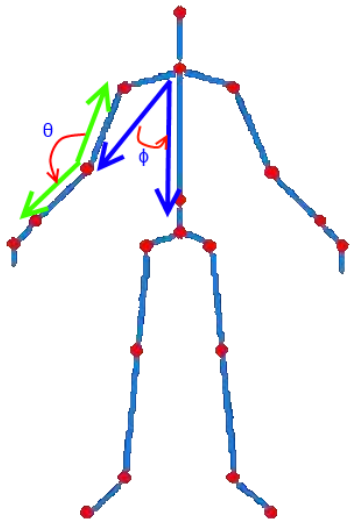

FIGURE 2.2 – Exemples de coordonnées cartésiennes (points rouges) et angulaires des articulations (flèches rouges) fournies par les systèmes de capture.

une action est un motif (*pattern*) qui s'étale sur une certaine durée et comporte donc une propriété de séquencement des données. De ce point de vue, les approches peuvent être regroupées en deux grandes familles. D'une part, les approches dites **séquentielles** permettent de modéliser implicitement ce séquencement au moyen d'un ensemble d'états et de transitions. D'autre part, les approches dites **statistiques** proposent d'extraire un vecteur descripteur de taille fixe, où la dimension séquentielle y est explicitement intégrée. A l'opposée des approches séquentielles, les approches statistiques produisent toujours des représentations de même taille indépendamment de la longueur de l'action représentée. Cette distinction est illustrée dans la Figure 2.3.

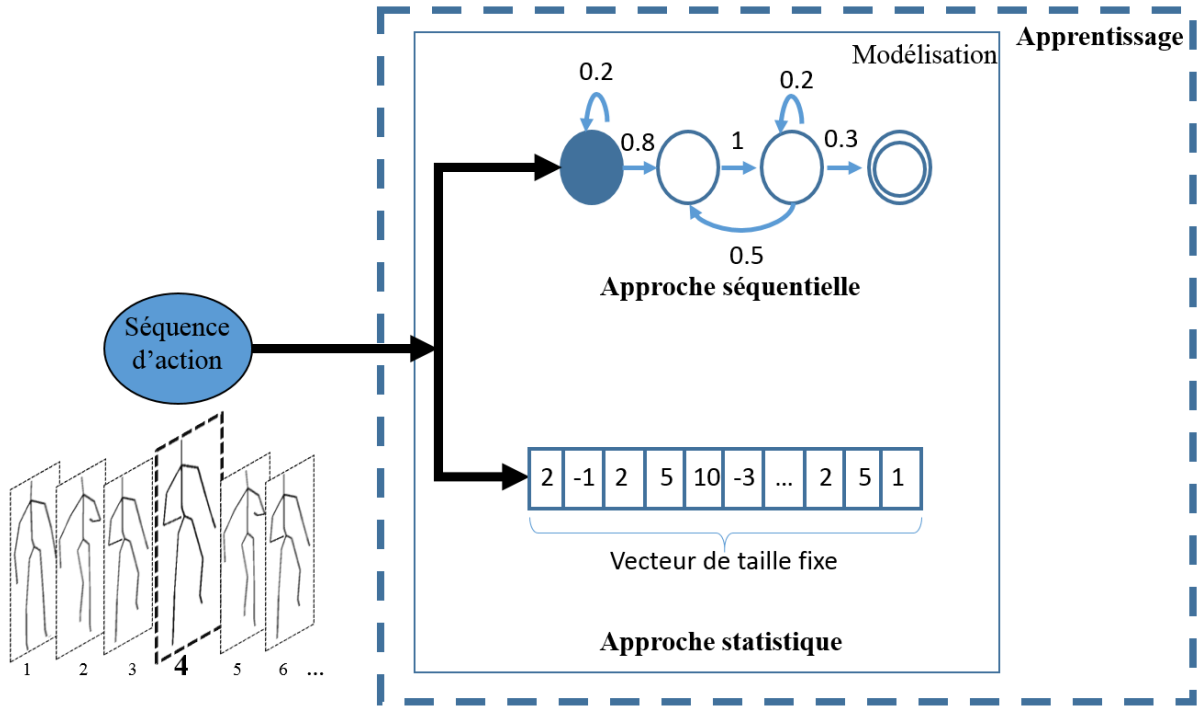

FIGURE 2.3 – Distinction entre une approche séquentielle et une approche statistique.

Le troisième critère différenciant les approches de reconnaissance d'actions 3D est la quantité de données nécessaire pour entraîner le modèle en question. En effet, dans le cas où très peu de données sont disponibles (une dizaine d'échantillons par classe), il est nécessaire de définir explicitement les descripteurs (la représentation) à extraire d'une action et ne garder au moyen d'une sélection que les plus pertinents. Nous qualifions ces approches de **transparentes** de par le fait que la constitution des représentations qui en résultent est parfaitement connue. Au contraire, dans le cas où le domaine d'application permet de disposer d'une grande quantité de données (des milliers d'échantillons par classe), il devient possible d'automatiser la tâche d'extraction et de sélection des descripteurs les plus à même de représenter une action. Cette deuxième catégorie d'ap-

proches est plus connue sous l'appellation d'approches à base **d'apprentissage profond**. Ainsi, il est important de connaitre la quantité de données à disposition dans le domaine considéré afin de choisir l'approche la plus adaptée. Comme illustré sur la Figure 2.4, les approches transparentes surpassent les approches à base d'apprentissage profond lorsque peu de données sont disponibles. Néanmoins au fur et à mesure que la quantité de données augmente, les approches transparentes tendent vers un état stationnaire alors que les approches à base d'apprentissage profond s'améliorent continuellement en présence de nouveaux échantillons d'apprentissage et arrivent à surpasser les approches transparentes.

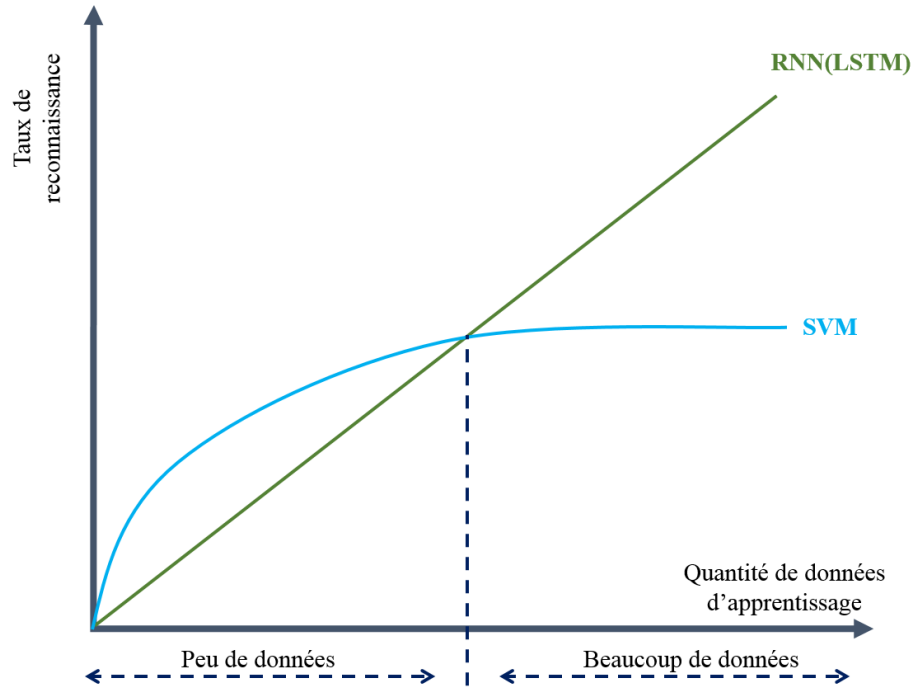

FIGURE 2.4 – Illustration des performances d'une approche transparente utilisant un SVM et une approche à base d'un apprentissage profond utilisant des LSTM.

Enfin, le dernier critère concerne la distinction entre les approches de reconnaissance d'actions dans un flot pré-segmenté et les approches dites de **détection** permettant la reconnaissance d'actions dans un flot continu (non segmenté). Ainsi, il est plus aisé de reconnaître une action lorsque les instants de début et de fin sont connus au préalable. Contrairement à la reconnaissance d'actions, durant laquelle on classifie une action pré-segmentée, la détection d'actions est une tâche combinant la segmentation et la reconnaissance car opérée sur une séquence non segmentée. Le but étant de détecter si une action donnée se produit, et si c'est le cas, d'en déterminer la classe (reconnaissance) ainsi que le début et la fin (segmentation).

Le reste de ce chapitre comporte une étude détaillée des approches pour les quatre

critères développés plus haut. En particulier, nous présentons dans la section 2.2 les approches de reconnaissance d'actions suivant qu'elles se basent sur des coordonnées cartésiennes, des angles articulaires ou des relations géométriques. Dans la section 2.3, nous distinguons entre les approches séquentielles et les approches statistiques. Dans cette section nous décrivons séparément les approches de reconnaissance d'actions à base d'apprentissage profond comme étant des approches récentes ayant besoin de grandes quantités de données d'apprentissage. Dans la section 2.4, nous décrivons les principales approches de détection d'actions non segmentées.

2.2 Typologie des données d'entrée

Une première manière de distinguer les différentes approches pour représenter une action 3D est de les répertorier selon le type de données squelettiques utilisées comme données d'entrée. En effet, les données squelettiques peuvent consister en des coordonnées cartésiennes des articulations ou bien leurs angles, qui peuvent aussi bien être utilisées en tant que valeurs absolues ou relatives. D'autres travaux proposent de se baser sur des booléens traduisant les relations géométriques entre différentes parties du corps. En plus d'être utilisées séparément, ces différentes modalités peuvent être exploitées simultanément dans le cadre d'approches multimodales.

Nous proposons dans les sections qui suivent de décrire ces familles d'approches en mettant en exergue les contextes d'application de chacune. Mais avant d'entamer la présentation de ces approches, nous introduisons dans une première section les différentes techniques d'acquisition des données squelettiques.

2.2.1 Techniques d'acquisition des données squelettiques 3D

Plusieurs systèmes de capture de mouvements permettent aujourd'hui l'acquisition des données squelettiques 3D. Il y a ceux qui opèrent une acquisition directe des positions articulaires ce qui permet d'avoir des mesures très précises. A l'opposé, une autre famille de systèmes utilise des méthodes d'estimation des positions articulaires et de reconstruction du squelette 3D à partir des données optiques comme les images RGB et/ou des cartes de profondeurs.

2.2.1.1 Capture directe de mouvement 3D

Les systèmes de capture de mouvement (Mocap) identifient et suivent les marqueurs qui sont attachés aux articulations ou aux parties du corps d'un sujet humain pour obtenir des informations sur le squelette 3D. Il existe deux catégories principales de systèmes

MoCap : d'une part les systèmes basés sur des caméras et d'autre part les systèmes basés sur des centrales inertielles. Les premiers emploient plusieurs caméras positionnées autour d'un sujet pour suivre, dans l'espace 3D, des marqueurs réfléchissants attachés au corps humain (Figure 2.5). Au contraire, dans les systèmes MoCap basés sur des capteurs inertiels, chaque capteur inertiel ayant 3 axes estime la rotation d'une partie du corps par rapport à un point fixe. Cette information est recueillie pour obtenir les données du squelette sans aucun dispositif optique autour d'un sujet. Les logiciels pour collecter des données squelettiques sont fournis avec des systèmes MoCap commerciaux, tels que Nexus pour Vicon [Vic18], Motive pour OptiTrack [Opt18], etc. Les systèmes MoCap, notamment basés sur plusieurs caméras, peuvent fournir des informations squelettiques 3D très précises à très haute vitesse. Néanmoins, de tels systèmes sont d'une part très coûteux et d'autre part nécessitent une certaine durée d'équipement du sujet.

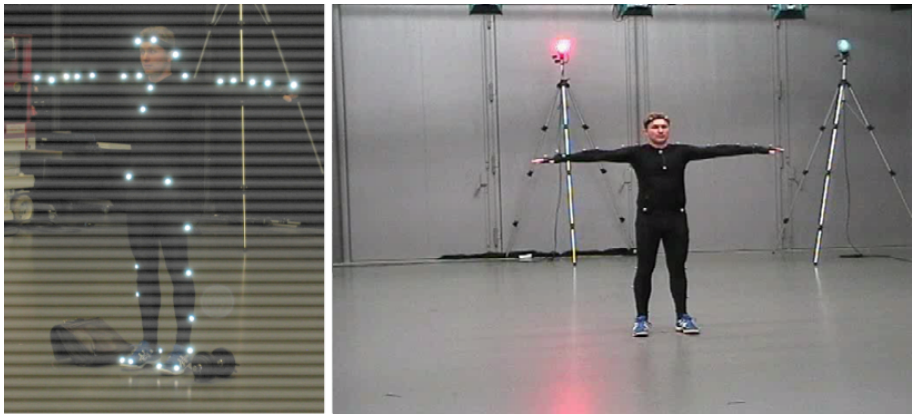

FIGURE 2.5 – Système de capture de mouvement optique basé sur des marqueurs rétroréfléchissants attachés au corps de l'acteur. Les marqueurs sont suivis par un ensemble de six à douze caméras haute résolution disposées en cercle [MRC+07].

2.2.1.2 Estimation de mouvement à partir d'images de profondeur

Grâce aux informations géométriques qu'une image de profondeur peut fournir, de nombreuses méthodes sont développées pour construire un modèle de squelette humain en 3D basé sur une seule image de profondeur ou une séquence de frames de profondeur. L'estimation des articulations humaines via la reconnaissance des parties du corps est une approche populaire pour construire le modèle du squelette [SSK+13, GSK+11, YJLSHDY15, SKS12, CE11, HOCB11, PGKT10, SMMN12].

En 2011, Shotton et al. [SSK+13] ont fourni un algorithme de construction de squelette extrêmement efficace, basé sur la reconnaissance des parties du corps et utilisable en temps réel. Une image de profondeur unique (indépendante des images précédentes) est

classée par pixel au moyen de forêts d'arbres décisionnels. Chaque branche de la forêt est déterminée par une simple relation entre le pixel cible et plusieurs autres. Les pixels qui sont classés dans la même catégorie forment la partie du corps, et l'articulation est déduite par la méthode de décalage moyen d'une certaine partie du corps, en utilisant les données de profondeur pour les déplacer dans la silhouette. Alors que la formation des forêts de décision nécessite un grand nombre d'images (environ 1 million) et une puissance de calcul considérable, le fait que les branches dans la forêt soient très simples permet à cet algorithme de générer des modèles 3D de squelette humain en environ 5 ms. La caméra Kinect de Microsoft est un des systèmes de capture se basant sur l'algorithme proposé par Shotton et al. [SSK+13] (Figure 2.6).

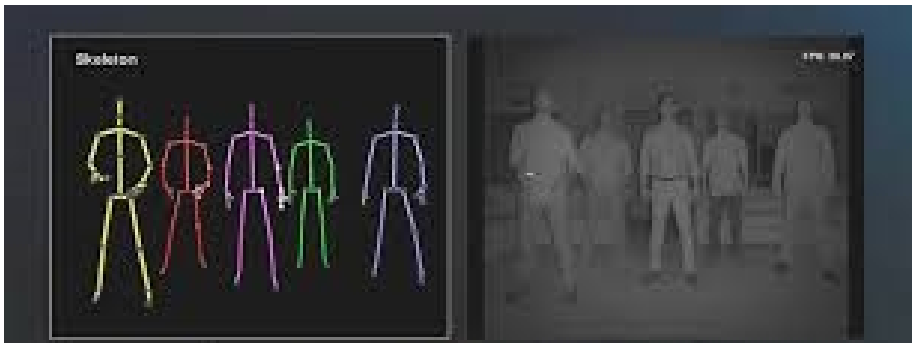

FIGURE 2.6 – Illustration de capture de données squelettiques à base de Kinect [Bre18].

Nous entamons à présent la description des approches suivant qu'elles utilisent les coordonnées cartésiennes (absolues ou relatives), les coordonnées angulaires (absolues ou relatives), les relations géométriques ou plusieurs modalités.

2.2.2 Coordonnées cartésiennes des articulations

Les coordonnées cartésiennes des articulations sont le type de données d'entrée le plus utilisé par les approches squelettiques de reconnaissance d'actions 3D. Ces données correspondent aux positions (x, y, z) des différentes articulations dans un repère cartésien souvent centré sur le système de capture. Ces positions permettent de reconstituer les trajectoires de chacune des articulations du squelette lors de la performance d'une action donnée. Ces trajectoires sont alors différemment exploitées pour modéliser une action tout en proposant de réduire au minimum la dépendance aux sujets ayant effectué l'action. En particulier, il est possible d'identifier deux sous-familles d'approches : celles utilisant les coordonnées absolues des articulations et celles basées sur leurs positions relatives.

2.2.2.1 Coordonnées cartésiennes absolues

Plusieurs approches exploitent directement les positions cartésiennes absolues des articulations. Il s'agit en fait des données brutes telles que fournies par les systèmes de capture mais qui sont souvent transformées de ce repère à un repère centré sur le sujet ayant pour origine le centre articulaire de la hanche.

Il existe plusieurs variétés de représentations qui exploitent les coordonnées cartésiennes absolues. Il est possible d'abord de citer les approches qui conçoivent une représentation sur la base des **trajectoires** issues des coordonnées cartésiennes absolues [WZZZ13b, GMLW14]. D'autres approches proposent un **changement d'espace de représentation** de manière à être plus robustes aux problèmes de variation d'angles de vue et des morphologies des sujets, tout en maintenant la structure intiale du squelette [VAC14, ESH14]. Les approches basées sur des **procédures de comptage** utilisent aussi les données absolues car plus informatives que les déplacements relatifs spatiaux ou temporels [HTGES13, COK$^+$13]. Plus récemment, les représentations basées sur un **apprentissage profond** utilisent les positions squelettiques brutes où le modèle recherche automatiquement les relations spatio-temporelles dans ces données, de la même façon que des techniques d'apprentissage profond extraient des caractéristiques sur des images de pixels bruts. Nous nous focalisons dans cette section sur les représentations autres que celles à base d'apprentissage profond vu qu'elles sont adressées de façon détaillée dans la section 2.3.3.

D'abord, en ce qui concerne les approches qui utilisent les positions absolues des articulations pour former une **trajectoire**, un exemple est donné avec l'approche proposée par Wei et al. [WZZZ13b]. Cette approche reçoit en entrée une séquence de poses squelettiques 3D. Chaque pose comporte les coordonnées cartésiennes 3D de K articulations. Comme illustré sur la Figure 2.7, cette séquence permet alors de former K trajectoires, chacune rapportant la progression d'une articulation donnée. En considérant chacune des K trajectoires comme un signal tridimensionnel, les auteurs proposent de les décomposer au moyen de la transformée en ondelettes. Ceci consiste en fait à calculer l'ensemble des produits scalaires de chaque trajectoire 3D avec des fonctions prédéfinies, présentant certaines propriétés mathématiques comme être oscillantes et de moyenne nulle. Ces fonctions sont appelées des ondelettes, les nombres obtenus sont appelés coefficients d'ondelettes et l'opération de détermination de ces coefficients est dite transformée en ondelette. Pour constituer leur représentation, les auteurs proposent de ne retenir pour toute trajectoire k que les V premiers coefficients (V varie en fonction des datasets), donnant lieu au vecteur H_k. Ainsi, la représentation finale x de la séquence considérée (une action) est une concaténation des vecteurs H_k, c'est-à-dire $x = [H_1, H_2, ..., H_K]$.

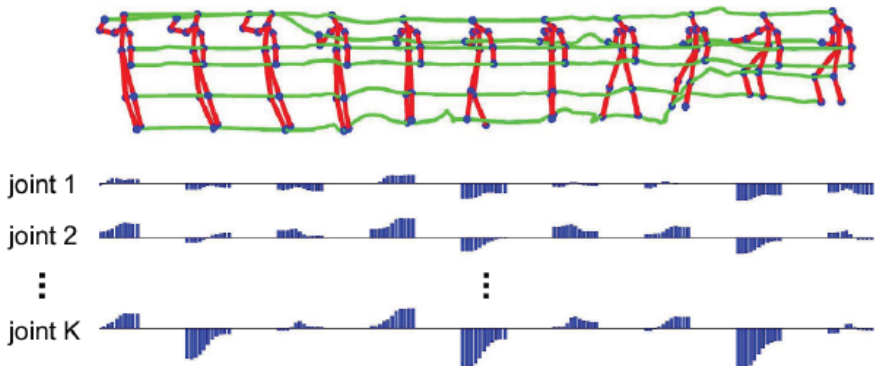

FIGURE 2.7 – Représentation composée de descripteurs d'ondelettes extraits à partir des trajectoires 3D [WZZZ13b].

Un autre exemple important des représentations se basant sur les coordonnées cartésiennes brutes est celui proposé par Evangelidis et al. [ESH14]. Cette représentation, dénommée *skeletal quad*, est issue de l'extraction de descripteurs à partir de chaque quadruplet articulaire (par exemple le quadruplet formé par les articulations du bras gauche dans la Figure 2.8). L'extraction des descripteurs est en réalité opérée avec les nouvelles coordonnées du quadruple articulaire exprimées dans un repère local. En particulier, si on considère un quadruple articulaire $J = [j_1, j_2, j_3, j_4]$ de telle sorte que (j_1, j_2) est la paire des articulations les plus éloignées du quadruple considéré, on définit un repère local tel que j_1 en est l'origine et j_2 est mappé sur $[1, 1, 1]^T$. Le changement de repère associé peut alors être défini par une matrice de transformation P (une matrice de rotation, un vecteur de translation et un facteur d'échelle) dont les paramètres sont déterminés à travers les données absolues de j_1 et j_2 et les contraintes $P(j_1) = [0, 0, 0]^T$ et $P(j_2) = [1, 1, 1]^T$. Enfin

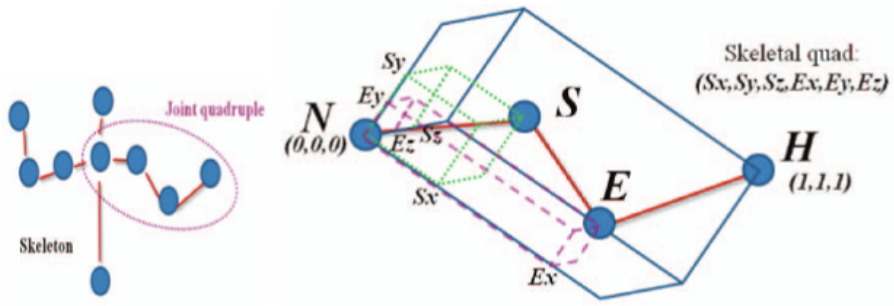

FIGURE 2.8 – Un exemple de codage d'un quadruple articulaire composé du [Cou, Epaule, Coude, Main]. Les articulations du cou et de la main correspondent aux points (0,0,0) et (1,1,1) dans le nouveau système de coordonnées locales. Les coordonnées 3D locales des articulations de l'épaule et du coude décrivent la structure du quadruple [ESH14].

le quadruple articulaire peut alors être encodé uniquement avec six paramètres, à savoir $q = [P(j_3); P(j_4)]$, appelé pour cette raison *skeletal quad*. Cette procédure est répétée pour K quadruples sélectionnés pour caractériser une action. L'avantage de cette approche est qu'elle offre une technique de normalisation basées sur les informations locales. Néanmoins les informations des articulations extrêmes sont en partie perdues car elles sont toujours mises à des valeurs prédéfinies.

Une autre approche intéressante proposée par Vemulapalli et al. [VAC14] et se basant sur les coordonnées cartésiennes brutes fait intervenir en même temps la notion de trajectoire et de changement d'espace de représentation. Comme illustrée dans la Figure 2.9, l'idée principale de cette approche est d'exprimer une séquence dans un nouvel espace, dit **espace courbe**, plus vaste que celui fourni par l'espace euclidien dans lequel une telle séquence est initialement exprimée. En particulier, l'espace courbe dans lequel est exprimée la nouvelle représentation de la séquence est dénommé le groupe de Lie (*Lie group*). L'avantage de ce nouvel espace est la possibilité de déterminer plusieurs types d'invariants qui permettent de caractériser la topologie des objets représentés.

Pour ce qui est des approches comptabilisant les positions cartésiennes absolues, il est possible de citer la représentation **Cov3DJ** proposée par Hussein et al. [HTGES13]. Cette représentation se base sur le calcul de la covariance statistique entre les valeurs absolues des positions articulaires. Plus de détails sur les descripteurs résultants sont donnés dans la section 2.3.2.2 portant sur la modélisation et la classification.

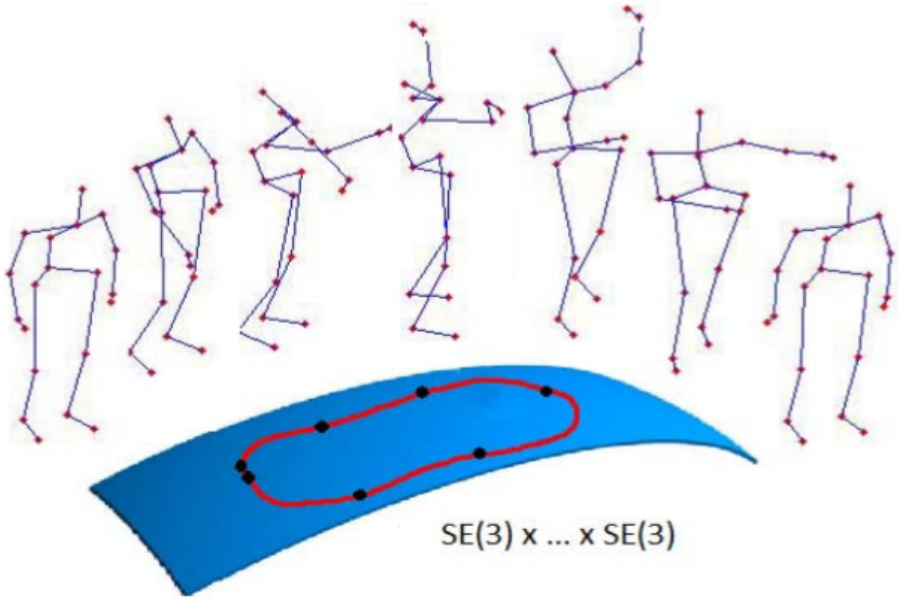

FIGURE 2.9 – Illustration des trajectoires 3D représentées comme une courbe dans le groupe de Lie suite au changement de l'espace de représentation [VAC14].

2.2.2.2 Coordonnées cartésiennes relatives

Comme évoqué dans plusieurs études [PLC16, HRHZ17], une grande partie des approches de reconnaissance d'actions squelettiques se basent sur le calcul des positions relatives entre les articulations. Ces approches néanmoins peuvent procéder selon deux manières. D'une part, certaines approches, comme celles introduites par [WLWY12, WLW14], proposent de calculer la différence d'emplacements $p_{ij} = p_i - p_j$ de chaque paire d'articulations (i, j) et cela à chaque instant. L'hypothèse de ces approches est qu'une action peut être caractérisée par les différences spatiales entre les articulations formant le squelette. D'autre part, il existe des approches [EMT+13, WZZZ13a, RMHM14] qui se basent sur le calcul de la différence d'emplacement d'une même articulation à différents moments sur une séquence de frames. Ces approches considèrent au contraire très peu l'information spatiale mais focalisent leur attention sur la progression temporelle des articulations séparément. Plus récemment, des approches proposent de combiner les avantages des deux techniques en intégrant dans leur représentations deux ou plusieurs variantes de ces descripteurs. Pour illustrer simultanément les deux types d'informations relatives, considérons la représentation introduite par Yang and Tian [YT12, YT14] dénommée *Eigen-Joints* ou articulations propres et qui fait partie des approches combinant les déplacements relatifs spatiaux et temporels.

Cette représentation, illustrée sur la Figure 2.10, comprend trois types de données

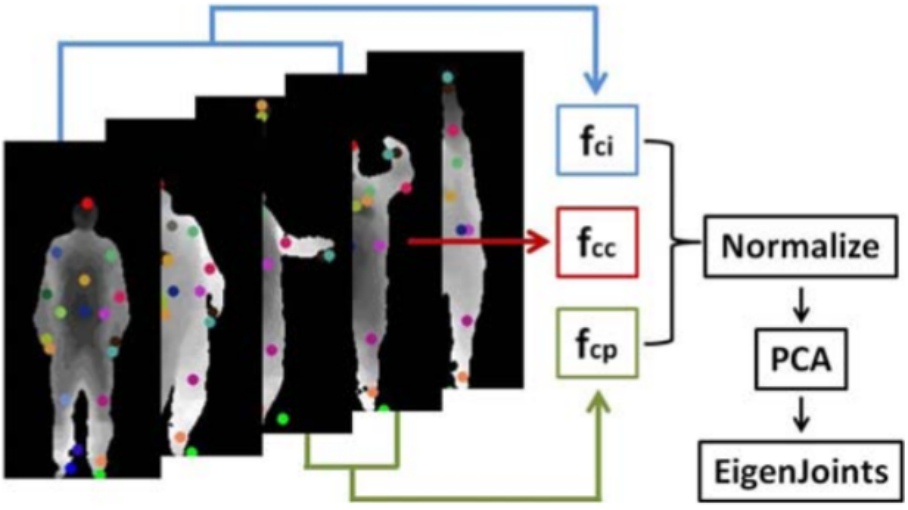

FIGURE 2.10 – Schématisation du procédé d'extraction des *Eigen-Joints*. Les auteurs se basent sur trois types de données relatives pour chaque frame dont la position relative dans une même frame f_{cc}, la position relative entre deux frames f_{cp} et la position relative par rapport à la frame initiale f_{ci} [YT12].

d'entrée. Il s'agit de déplacements relatifs calculés à chaque frame c dont : la position relative dans une même frame f_{cc}, la position relative entre deux frames f_{cp} et la position relative par rapport à la frame initiale f_{ci}. Ces valeurs sont ensuite concaténées pour générer un seul vecteur descripteur $f_c = [f_{cc}, f_{cp}, f_{ci}]$ pour chaque frame c.

En particulier, étant données les coordonnées cartésiennes 3D $X = x_1^c, x_2^c, ..., x_N^c$ de N articulations, le déplacement de posture f_{cc} permet de caractériser l'information spatiale d'une posture à l'instant c. Ce déplacement est calculé comme suit :

$$f_{cc} = \{x_i^c - x_j^c | i, j = 1, 2, ..., N; i \neq j\} \tag{2.1}$$

Pour capturer la propriété de mouvement de la frame courante c de façon locale, les différences entre toutes les paires d'articulations sont calculées entre la frame courante c et la frame précédente p comme suit :

$$f_{cc} = \{x_i^c - x_j^p \quad | \quad x_i^c \in X_c \quad x_j^p \in X_p\} \tag{2.2}$$

Où X_c et X_p représentent, respectivement, l'ensemble des articulations de la frame courante c et la frame précédente p.

Enfin pour caractériser le déplacement global de la frame courante c, les différences entre toutes les paires d'articulations sont calculées entre la frame courante c et la frame initiale I, comme suit :

$$f_{cc} = \{x_i^c - x_j^I \quad | \quad x_i^c \in X_c \quad x_j^I \in X_I\} \tag{2.3}$$

Le point le plus intéressant de cette représentation est le fait qu'elle propose d'inclure des informations spatiales (déplacement relatif au sein d'une même frame), des informations temporelles locales (déplacement entre deux frames) et des informations temporelles globales (déplacement global). Il est en effet important de ne pas se limiter seulement aux informations de posture ou se baser uniquement sur l'étude de la dynamique du mouvement, car une action est un pattern complexe où spatialité et temporalité sont des informations cruciales pour la discriminer. Un autre point intéressant de cette représentation est la normalisation des valeurs qui sont maintenues dans un même intervalle (dans ce cas dans l'intervalle [-1,+1]). Ceci permet notamment de faire face à des problématiques telles que la variabilité de la morphologie des sujets induisant des déplacements très distincts et donc une difficulté de caractériser une classe d'action.

Par contre, nous pensons que considérer les données issues de l'intégralité des paires d'articulations, comme proposé dans cette approche, peut impacter négativement la puissance de la représentation notamment à cause de sa très grande dimensionnalité. Des réductions sont alors nécessaires et peuvent conduire à des représentations ne tenant pas

compte de tous les aspects spatio-temporels d'une action. Plus encore, il est possible de noter qu'au moment où plusieurs descripteurs extraits sont relatifs au même aspect spatial ou temporel, plusieurs autres aspects ne sont pas considérés. A titre d'exemple, en considérant uniquement la frame précédente, la relation temporelle d'une même articulation est très faiblement modélisée.

Ainsi, nous retenons d'une part l'intérêt d'extraire à la fois les informations spatiales et temporelles mais aussi l'intérêt d'intégrer une étape de normalisation des données pour adresser les différences morphologiques des sujets. D'autre part, nous relevons que plusieurs approches extraient seulement une partie des informations spatio-temporelles et qu'il faudrait s'assurer que la représentation finale correspond aux aspects spatio-temporelles qu'on veut initialement modéliser en évitant d'en supprimer certains suite à une réduction ou autre opération de simplification.

2.2.3 Angles articulaires

Une autre modalité largement utilisée pour la construction de représentations d'actions 3D est l'orientation articulaire. Se baser sur cette modalité permet en général de former des caractéristiques invariantes par rapport à la position du sujet, la taille du corps et l'orientation de la caméra. Comme pour la section portant sur les coordonnées cartésiennes, il est possible, à ce niveau, de distinguer aussi entre les approches utilisant en entrée des angles articulaires relatifs ou absolus. Nous abordons ci-après les deux sous-catégories de familles d'approches, en détaillant, à chaque fois, quelques exemples.

2.2.3.1 Angles articulaires absolus

Il s'agit des représentations qui sont basées sur les valeurs brutes des angles d'orientation de chaque articulation. L'angle d'une articulation est généralement exprimé par rapport à l'articulation parent à laquelle elle est reliée. La Figure 2.11a illustre la structure du squelette tel que fourni dans la base d'actions **Berkeley MHAD** [OCK+14] ainsi que les 21 angles articulaires.

La représentation proposée par Ofli et al. [OCK+14] dénommée **SMIJ** est un exemple intéressant d'approches basées sur les angles articulaires absolus. En particulier, pour un squelette de J articulations, les données d'entrée correspondant à une séquence d'action s'étalant sur T frames sont l'ensemble $A = [a^1 a^2 ... a^J]$ de J séries temporelles d'angles, où $a^j = \{a_t^j\}_{t=1}^{t=T}$ pour $1 \leq j \leq J$ et a_t^j est l'angle à l'instant t associé à l'articulation j. Pour ces auteurs, la principale motivation d'utiliser les angles articulaires est d'assurer une certaine invariance aussi bien par rapport à la morphologie des sujets que par rapport au système de coordonnées adopté. En outre, l'hypothèse sur laquelle repose la représen-

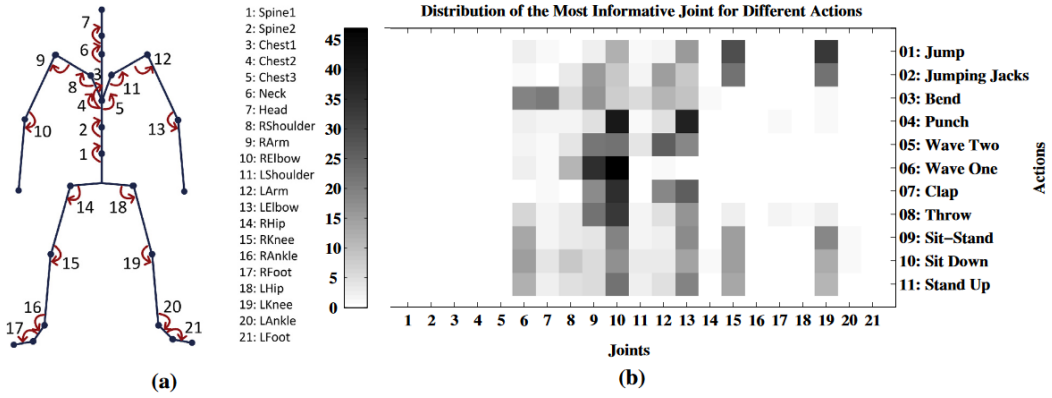

FIGURE 2.11 – (a) Illustration des 21 angles articulaires bruts tels que fournis dans la base MHAD. (b) Distribution des articulations les plus informatives pour différentes classes d'actions de la base MHAD. Chaque entrée correspond au pourcentage du temps où une articulation donnée est considérée comme la plus informative pour une action donnée (plus sombre signifie un pourcentage plus élevé) [OCK+14].

tation proposée est que différentes actions exigent que les sujets activent des articulations différentes et d'une manière différente. La Figure 2.11b permet de visualiser si une articulation s'active ou non et à quelle intensité lors de la performance d'une action donnée de la base Berkeley MHAD. Sur la base de cette constatation, la représentation résultante est une séquence ordonnée (suivant l'intensité d'activation) des angles des N (dépendant de la base d'actions) articulations les plus informatives à chaque instant.

2.2.3.2 Angles articulaires relatifs

Comme pour les représentations basées sur les positions cartésiennes relatives, il est possible de distinguer les approches spatiales, qui se basent sur les orientations des articulations au même instant (à la même frame), des approches dites temporelles, calculant la différence entre les orientations d'une même articulation à travers une séquence de frames.

Une des représentations spatiales les plus répandues consiste à calculer l'orientation de chaque articulation par rapport au centre articulaire du squelette. Par exemple, Gu et al. [GDOS12] ont collecté les données de quinze articulations et ont extrait comme descripteurs les angles articulaires par rapport au centre articulaire du torse. Sung et al. [SPSS12] ont calculé une matrice d'orientation pour chacune des articulations par rapport à la caméra, puis ont transformé la matrice de rotation articulaire pour obtenir l'orientation de l'articulation par rapport au centre articulaire du torse. Une approche similaire a été introduite dans [SPSS11] en se basant également sur une matrice d'orientation.

Une autre approche spatiale consiste à calculer l'orientation entre chaque paire d'arti-

culations, appelées orientations relatives des articulations. Par exemple, Zhang and Tian [ZT12] ont utilisé une représentation squelette 3D combinant des données structurelles avec des données de mouvement. Les données structurelles consistent en des descripteurs reliant les positions de chaque paire d'articulations les unes par rapport aux autres. L'orientation entre deux articulations i et j a également été utilisée. Elle est donnée par $\Theta(i,j) = \arcsin(\frac{i_x - j_x}{dist(i,j)})/2\pi$ où $dist(i,j)$ désigne la distance géométrique entre deux articulations i et j dans l'espace 3D.

Dans la catégorie des représentations temporelles, on retrouve par exemple l'approche proposée par Campbell and Bobick [CB95] qui établit une transformation de l'espace cartésien à l'espace des phases, un espace abstrait dont les coordonnées sont les variables dynamiques (angles articulaires) de l'objet à représenter (une action). Dans cet espace, il est possible de visualiser facilement la progression d'une action sous forme d'une courbe qui peut être comparée à d'autres courbes de mouvement. Une autre représentation proposée par Boubou and Suzuki [BS15] est basée sur un Histogramme des Vélocités Orientées (HOVV), qui est un histogramme des vitesses angulaires calculées pour 19 articulations. Il s'agit d'une autre technique qui permet de considérer l'information angulaire relative sous forme de vitesses.

2.2.4 Relations géométriques

Au lieu d'utiliser les différentes mesures qu'il est possible de collecter pour une articulation (position, vitesse, angles, etc.), les méthodes de cette catégorie tentent de représenter une action au moyen des relations géométriques entre différentes parties du squelette. Un des travaux caractéristiques de cette famille d'approches est la représentation proposée par Muller et al. [MRC05]. Cette représentation se base sur un ensemble de booléens indiquant la disposition des différentes articulations les unes par rapport aux autres (par exemple la jambe gauche devant la jambe droite, la tête devant le torse, etc.).

Sur la Figure 2.12a, il est possible de relever par exemple que lors de la performance de l'action "marcher" les jambes occupent des emplacements opposés. Ainsi, au lieu d'avoir en entrée les positions absolues ou relatives des différentes articulations, les données d'entrée seraient des vecteurs booléens caractérisant les relations d'emplacement des articulations de façon à la fois simple (que des 0 et 1) et robuste (même si certaines positions articulaires sont bruitées). Les auteurs proposent au moyen de ces données de définir des descripteurs de différents niveaux de complexité. En effet, il est possible de définir des fonctions dites **booléennes** qui prennent en paramètre plusieurs articulations ou d'autres fonctions booléennes et rendent le résultat (0 ou 1) d'opérations appliquées sur ces paramètres telles que la multiplication booléenne.

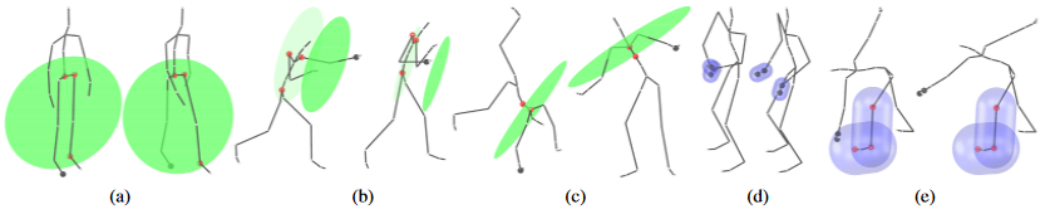

FIGURE 2.12 – Illustration des données décrivant les relations géométriques entre les différentes articulations du corps, indiquées par des marqueurs rouges et noirs, au niveau d'une même pose [MRC05].

Plus récemment, et en plus des relations géométriques entre les articulations, Zhang et al. [ZYX+18, ZLX17] proposent de considérer comme données d'entrée les relations géométriques entre des plans articulaires définis par au moins trois articulations. Comme illustré sur la Figure 2.13, les auteurs proposent néanmoins de ne pas considérer tous les plans articulaires, mais sélectionner ceux dont les dispositions respectives sont pertinentes pour représenter une action.

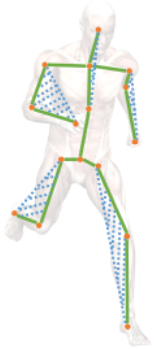

FIGURE 2.13 – Illustration des plans articulaires à considérer pour exprimer les relations géométriques permettant de caractériser une action [ZYX+18].

2.2.5 Multimodalité

Puisque plusieurs modalités d'informations sont disponibles, une façon intuitive d'améliorer le pouvoir descriptif d'une représentation est d'intégrer plusieurs sources d'informations et de construire une représentation multimodale pour encoder les actions 3D. D'une part, certains travaux proposent d'intégrer ensemble les angles et les positions cartésiennes d'une articulation pour construire des représentations [GFA06, GTHES13, YLY14, PDLM15]. Bien qu'étant multimodales, ces approches ne combinent que des don-

nées de nature squelettique. D'autre part, des travaux ont examiné l'intérêt d'exploiter aussi bien les données squelettiques que des images RGB ou des cartes de profondeurs [ZCG13, CJK16, WLWY12, WLW14]. Nous illustrons ces deux sous-familles en décrivant plus en détail un exemple de chacune d'entre elles.

Afin d'illustrer la première sous-famille d'approches multimodales, nous nous intéressons à la représentation proposée par Pazhoumand-Dar et al. [PDLM15]. Cette représentation n'est basée que sur des données squelettiques mais combinent les angles et les positions cartésiennes des articulations.

En particulier, les auteurs proposent d'abord de calculer pour chaque articulation l'angle φ entre le vecteur position de cette articulation et l'axe Y et l'angle θ entre la projection de ce vecteur sur le plan $(X-Z)$ et l'axe X. Il en résulte, pour chaque articulation $J_i, 1 \leq i \leq K$, une séquence d'angles $D(J_i) = \{(\varphi^t, \theta^t)\}_{t=1}^{t=T}$, où T et la longueur de la séquence et K le nombre total d'articulations. Comme pour la représentation SMIJ proposée dans [OCK+14], les auteurs soutiennent l'idée que des actions différentes activent des articulations différentes. De ce fait, la séquence d'angles $D(J_i)$ n'est calculée que pour les articulations ayant une importante contribution à la performance de la classe d'action considérée. Les auteurs proposent alors d'évaluer l'entropie de chaque articulation pour déterminer le niveau d'activation de chaque articulation. Un seuil par classe d'action est alors expérimentalement fixé de manière à ne retenir que les articulations dont la valeur de l'entropie est supérieure à ce seuil pour représenter la classe d'action associée. Ceci permet alors de former la séquence d'angles dénommée **MIJA** pour *most informative sequences of joint angles*.

De plus, ces auteurs considèrent les positions cartésiennes relatives entre plusieurs paires d'articulations. Comme pour former les MIJA, une sélection des paires d'articulations les plus informatives est aussi opérée. Néanmoins, cette sélection n'est pas basée sur une technique de seuillage mais sur une comparaison entre les concepts de variation intra-classe et inter-classes. En particulier, pour chaque paire d'articulations (J_i, J_j), une séquence notée SRP et composée des distances relatives entre ces deux articulations est calculée comme suit :

$$SRP(J_i, J_j) = \{\| \overrightarrow{f}^t_{J_i} - \overrightarrow{f}^t_{J_j} \|\}_{t=1}^{t=T} \tag{2.4}$$

Où $\overrightarrow{f}^t_{J_i}$ et $\overrightarrow{f}^t_{J_j}$ sont les vecteurs position à l'instant t des articulations J_i et J_j, respectivement.

La Figure 2.14 comporte deux illustrations des valeurs de $SRP(J_{Maingauche}, J_{Maindroite})$ pour deux instances d'une même classe d'action effectuées par deux sujets différents. Il est alors possible de relever que, d'une part, les séquences appartenant à des classes d'action différentes sont aisément distinguables, et que d'autre part, des instances appartenant

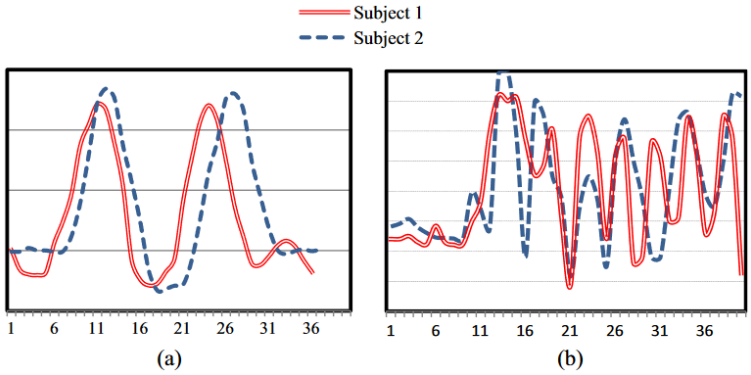

FIGURE 2.14 – Illustration des valeurs $SRP(J_{Maingauche}, J_{Maindroite})$ lors de la performance des actions (a) applaudissement avec les deux mains, (b) un service de tennis, effectuées par deux sujets. L'échelle horizontale affiche le numéro de la frame et l'échelle verticale indique la distance relative [PDLM15].

à une même classe d'action présentent des caractéristiques similaires bien que réalisées par des sujets différents. En se basant sur ces deux aspects, les auteurs proposent de retenir une paire d'articulations (J_i, J_j) pour servir à modéliser une classe d'action C si la variation intra-classe $V_{intra}(J_i, J_j)$ est supérieure à la moyenne des variations inter-classes $V_{inter}(J_i, J_j)$. Pour ce faire, la moyenne $V_{intra}(J_i, J_j)$ de la variation intra-classe de la séquence $SRP(J_i, J_j)$ est calculée pour l'ensemble des instances appartenant à la même classe C. De même, la variation inter-classes $V_{inter}(J_i, J_j)$ relative à la classe C est calculée entre les instances de cette classe et toutes les autres classes. Enfin, les paires d'articulations retenues forment un ensemble dits **MIRM** pour *most informative relative motions of joints*.

S'agissant de la deuxième sous-famille des approches multimodales, c'est-à-dire basée d'une part sur du squelette et d'autre part sur des images RGB ou carte de profondeur, une représentation intéressante est celle proposée par Wang et al. [WLWY12, WLW14]. Cette représentation comporte deux types d'information : (1) les positions relatives p_{ij} entre toutes les combinaisons possibles de paires d'articulations (i, j) et (2) un descripteur dit **LOP** (pour *Local Occupancy Patterns*) qui permet de décrire "l'apparence locale" autour des articulations. Pour constituer ce deuxième descripteur, les auteurs utilisent le nuage de points générés à chaque instant à partir de la carte de profondeur. Ainsi, pour chaque articulation J, la région locale autour d'elle est partitionnée en une grille de taille $N_x \times N_y \times N_z$. Chaque élément (bin) de la grille est composé de (S_x, S_y, S_z) pixels. L'idée est de comptabiliser dans un histogramme l'ensemble des points appartenant à chacun de ces bins au fur est à mesure que l'action progresse. Le descripteur **LOP** est alors formé suite à la concaténation de ces histogrammes (Figure 2.15).

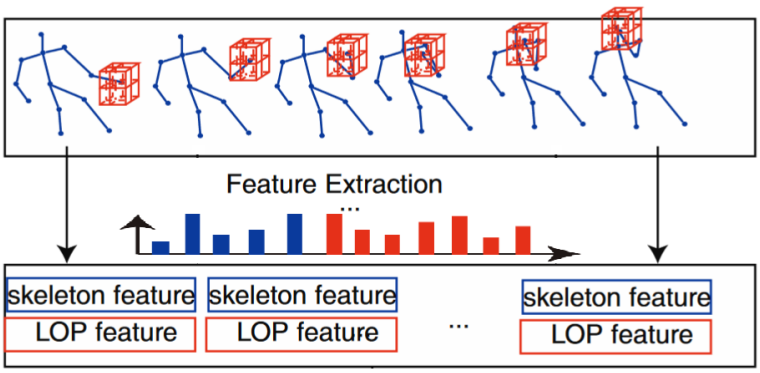

FIGURE 2.15 – Illustration des modalités de squelette et de profondeur utilisées pour construire la représentation dite **actionlet** [WLW14].

2.2.6 Discussion

Nous nous sommes penchés dans cette section sur le premier maillon du processus de reconnaissance d'actions, présenté dans la Figure 2.1. Il s'agit du type de données reçues en entrée par le système de reconnaissance et qui conditionne grandement le choix des techniques utilisées par la suite pour représenter une action et la reconnaître. En particulier, notre étude s'est focalisée sur les approches basées sur des données squelettiques. Ces données consistent soit en des coordonnées cartésiennes 3D (absolues ou relatives), des coordonnées angulaires (absolues ou relatives) ou bien des relations géométriques entre les différentes parties du corps. Certains travaux proposent même de combiner plusieurs modalités comme les coordonnées cartésiennes et angulaires ou encore des données squelettiques avec des images RGB ou des cartes de profondeur.

L'utilisation d'une modalité donnée se justifie par au moins deux critères. Le premier est la pertinence de la modalité pour mettre en évidence les traits saillants des actions à modéliser dans le cadre applicatif considéré. Le second critère est le degré de complexité des problèmes inhérents à la modalité utilisée et qui nécessitent d'être adressés explicitement. Par exemple, dans un cadre applicatif où les actions à reconnaître sont sensiblement distinctes, et sont donc plus facilement modélisables, il suffit d'utiliser des coordonnées angulaires ou des relations géométriques, qui rapportent moins d'informations que les coordonnées cartésiennes mais présentent moins de problèmes de variabilité morphologique des sujets. Au contraire, dans un cadre d'utilisation où les actions à reconnaître présentent beaucoup de similitude, il devient alors nécessaire de se baser sur des coordonnées cartésiennes, car plus complètes, mais il est également nécessaire d'adresser explicitement les problèmes relatifs aux variabilités morphologiques.

S'agissant maintenant du choix que nous avons fait pour mener nos travaux de recherche, et vu que l'objectif final est de concevoir une approche de reconnaissance d'ac-

tions dans un cadre d'interaction Homme-Machine où les classes d'actions peuvent être très similaires, nous nous basons sur des **coordonnées cartésiennes 3D absolues**. En particulier, nous considérons les trajectoires 3D de chacune des articulations squelettiques, et à partir desquelles plusieurs informations caractéristiques sont extraites. L'utilisation des trajectoires 3D est d'autant plus justifiée que notre objectif est de s'inspirer des approches de modélisation de tracés manuscrits 2D pour construire nos représentations. Néanmoins, il est important de souligner que notre approche intègre certaines informations angulaires mais sont pour la plupart déduites à partir des coordonnées cartésiennes et ne correspondent pas à des angles articulaires tels que introduits dans cette section. Enfin, ce choix nécessite d'identifier et d'adresser explicitement les difficultés majeures inhérentes à ce type de données comme par exemple le problème de variabilité morphologique évoqué plus haut.

2.3 Modélisation et classification des actions squelettiques 3D

Nous nous intéressons dans cette section au deuxième maillon du processus de reconnaissance d'actions 3D, illustré dans la Figure 2.1. En effet, après collecte et pré-traitement des données squelettiques, il faut procéder à la modélisation de l'action et à sa classification. Étant donné que l'action est un pattern ayant une propriété de séquencement (temporel), il est possible, comme pour d'autres domaines du traitement du signal tels que la reconnaissance de la parole ou de l'écriture, de répertorier les techniques de modélisation en deux familles. D'une part, nous retrouvons les approches dites **séquentielles** qui permettent de modéliser de manière implicite ce séquencement au moyen d'un ensemble d'états et de transitions ou bien par le biais d'une mesure de similarité. D'autre part, les approches dites **statistiques** proposent d'extraire un vecteur descripteur de taille fixe, où la dimension séquentielle y est explicitement intégrée. Plus récemment, des approches à base d'apprentissage profond ont été proposées pour modéliser et reconnaître des actions. Ces approches peuvent être de nature séquentielle (réseaux récurrents) ou statistique (réseaux de convolution) et ont la particularité de nécessiter une quantité importante de données d'apprentissage.

Ci-après nous présentons d'abord les approches séquentielles, ensuite les approches statistiques et enfin les approches à base d'apprentissage profond.

2.3.1 Approches séquentielles

Les premiers travaux s'inscrivant dans cette famille se sont basés soit sur des Modèles de Markov Cachés ou bien sur des comparaisons élastiques entre les séquences d'actions. Nous nous intéressons donc dans cette section à ces deux sous-familles d'approches séquentielles.

2.3.1.1 Modèle de Markov Caché

Un modèle de Markov caché (HMM pour *Hidden Markov Models*) est un automate qui représente chaque événement pouvant être observé sous forme d'un état. En optant pour un tel modèle, on fait l'hypothèse que la séquence modélisée, par exemple une action, est un processus possédant la propriété de Markov : l'information utile pour la prédiction de l'état futur (état i+1) est entièrement contenue dans l'état présent du processus (état i) et n'est pas dépendante des états antérieurs ($i-1, i-2, ...$). Ils sont un choix très commun pour reconnaître des actions, ce qui a donné lieu à plusieurs architectures de reconnaissance d'actions [WYW$^+$12b, GDOS12, XCL$^+$12, PL11, JW14, WYW$^+$12a, SGW$^+$13].

En pratique, on utilise deux types de modèles de Markov cachés, le modèle ergodique et le modèle gauche-droite. Le modèle ergodique est sans contrainte : toutes les transitions d'un état vers un autre sont possibles. Le modèle gauche-droite est un modèle partiellement connecté car contenant des contraintes. Dans le modèle gauche-droite le plus utilisé, l'état i n'est relié par une transition de probabilité non nulle qu'à trois états : lui-même, l'état i+1 et l'état i+2. Les modèles ergodiques peuvent modéliser un plus grand nombre de processus en tirant profit d'un plus grand nombre de paramètres, tandis que les modèles partiellement connectés tels que le modèle gauche-droite, sont plus simples (moins de paramètres) et nécessitent donc moins de données d'apprentissage [XCL$^+$12, FLO14].

Les HMM sont caractérisés par les composantes suivantes $\lambda = \pi, A, B, N, M$: N est le nombre d'états dans le modèle, M est le nombre de symboles distincts observés par état, π est la distribution d'état initiale, A est la matrice de probabilité de transition entre états et B est la matrice de probabilités d'émission ou d'observation des symboles dans chacun des états. La majorité des méthodes utilisent un seul HMM à N états pour chaque classe d'action. Le choix du nombre d'états N dépend de la complexité du processus à modéliser : un nombre élevé d'états pourrait générer un modèle trop spécifique ; de l'autre côté, avec un petit nombre d'états, il y a la possibilité d'avoir des actions indiscernables. Aucune règle générale n'existe dans la littérature pour résoudre ce problème. Un compromis approprié est généralement trouvé par validation expérimentale. A titre d'exemple, la Figure 2.16 illustre un HMM à cinq états avec les probabilités correspondantes, entraîné pour l'action "s'asseoir".

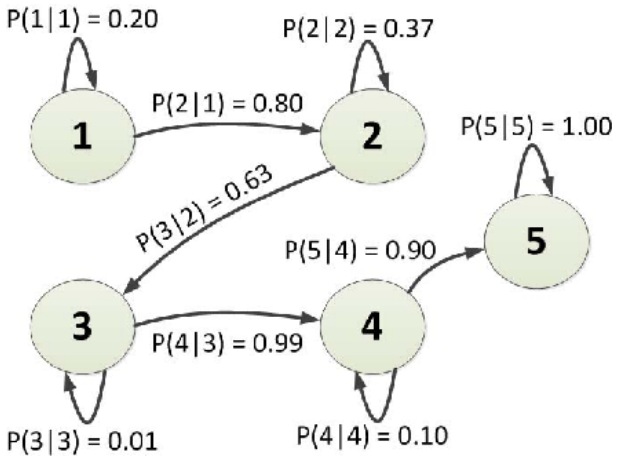

FIGURE 2.16 – Illustration d'un HMM gauche-droite à cinq états entraîné pour reconnaître l'action "s'asseoir" [GMM17].

Un des travaux importants ayant porté sur la reconnaissance d'actions squelettiques de façon globale et au moyen d'HMM de façon particulière, est celui mené par Xia et al. [XCA12]. Une vue d'ensemble de l'approche résultante est illustrée dans la Figure 2.17. En particulier, les auteurs proposent d'abord une nouvelle représentation appelée **HOJ3D**, qui permet de modéliser chaque posture (un squelette à chaque frame)

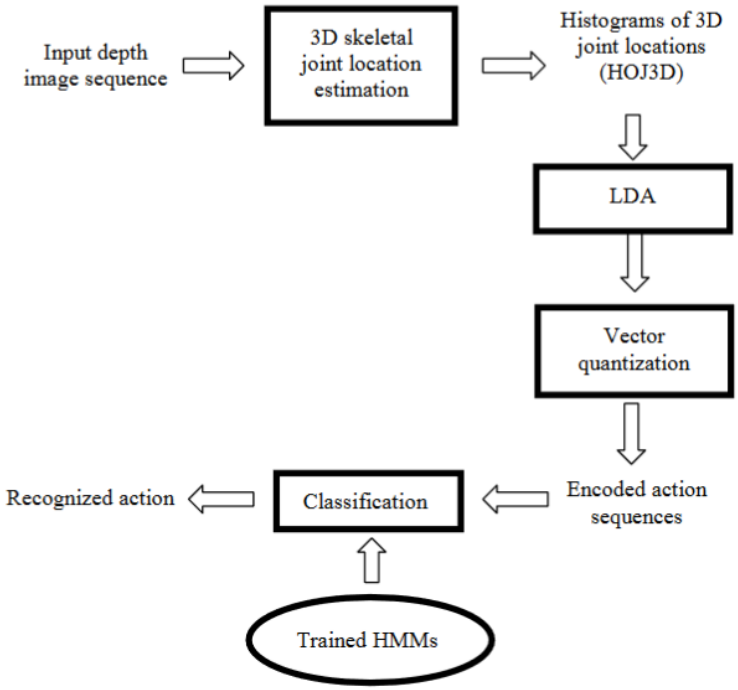

FIGURE 2.17 – Vue d'ensemble de la méthode proposée par Xia et al. [XCA12].

sous forme d'un histogramme de dimension $n = 84$. Cet histogramme comptabilise l'appartenance pondérée de neuf articulations sélectionnées à n régions qui compose une sphère entourant le sujet. Ensuite pour obtenir une représentation plus robuste, l'algorithme d'Analyse Discriminante Linéaire (**LDA** pour *Linear Discriminant Analysis*) est employé. Cet algorithme recherche les vecteurs dans l'espace sous-jacent pour obtenir la meilleure discrimination entre les différentes classes. A la suite de cette analyse, la dimension du vecteur HOJ3D par posture, initialement égale à n, est réduite pour être égale au nombre de classes d'actions à reconnaître. Enfin, pour entraîner le modèle HMM, les auteurs proposent de transformer chaque vecteur d'une posture en un symbole (une observation), pour que chaque action puisse être représentée par une séquence d'observations. Un ensemble d'observations est alors créé en regroupant toutes les postures en (K=125) clusters au moyen de l'algorithme de K-means. Chaque posture est ensuite représentée par un seul nombre, c'est-à-dire le numéro du cluster auquel elle appartient. Une action est alors représentée comme une série temporelle de nombres entiers. Un HMM discret à six états est alors entraîné pour chaque classe d'action au moyen de l'algorithme de Baum-Welch [BPSW70].

Afin de classifier une séquence d'action de test $V = \{v_1, v_2, ..., v_T\}$, il suffit de calculer via l'algorithme de Viterbi [Vit67] la probabilité de générer la séquence d'observations V sachant le modèle d'action HMM_i, c'est-à-dire $P(V|HMM_i)$, pour chaque modèle. La classe prédite correspond au modèle HMM_i ayant la probabilité maximale :

$$C^* = \arg\max_i \{P(V|HMM_i)\}, i = 1, 2, ..., M \tag{2.5}$$

où M est le nombre de classes d'actions.

Plusieurs autres approches se basant aussi sur des HMM ont été proposées. Par exemple, Xu et al. [XCL+12] proposent de contrôler un robot humanoïde en classifiant de manière continue les mouvements de l'utilisateur au moyen d'un dictionnaire gestuel de 6 mots, alors que Zhu et al. [ZP12] ont utilisé des HMM pour concevoir une plateforme interactive de reconnaissance d'actions dans laquelle l'utilisateur recevait des retours sur sa performance. Sorel et al. [Sor12] proposent une approche de reconnaissance des actions faites avec les bras où chaque classe d'action est modélisée par un HMM composé de 7 états. Wu et al. [WWCL13] proposent une architecture HMM pour détecter des gestes du langage des signes au moyen d'un dictionnaire de 20 mots.

Bien que le HMM représente un outil standard pour modéliser des patterns séquentiels tels que des actions, il présente quelques inconvénients qu'il faut prendre en compte. Un premier point concerne la définition de l'architecture du modèle qui inclut le choix du type du modèle (ergodique, gauche-droite ou autre), le nombre d'états, le choix des symboles d'observations (discrètes ou continues), etc. Il n'y a en effet aucune procédure théorique

pour déterminer ces composants, qui sont souvent fixés au terme de plusieurs validations expérimentales. Un deuxième point intéressant est le nombre important de paramètres à optimiser pour constituer le modèle, ce qui nécessite une quantité importante de données d'apprentissage. Or, pour certains domaines d'application, comme celui de l'interaction Homme-Machine auquel nous nous intéressons, il serait fastidieux de devoir collecter une quantité importante de données pour chaque nouvelle classe qu'un utilisateur voudrait intégrer à son système d'interaction.

2.3.1.2 Comparaison élastique

Une autre manière de considérer la nature séquentielle des patterns pour des fins de classification, est d'opérer un alignement entre une séquence de test et celles de référence. Ainsi, la classe de la séquence de test serait la même que celle de la séquence de référence à laquelle elle ressemble le plus, c'est-à-dire celle par rapport à laquelle elle est la moins éloignée. Pour cela, il faudrait définir une métrique de similarité permettant de mesurer l'éloignement par rapport à des séquences de référence.

Pour des séquences de même longueur, une procédure intuitive consisterait à comparer chaque couple de séquences directement au moyen de la distance euclidienne. Cela revient à calculer la différence entre des paires de points alignés de ces séquences et à les additionner. Cette distance est calculée entre la séquence de test à classifier et chaque séquence de référence de manière à trouver celle permettant d'avoir une distance minimale. Il se trouve qu'en pratique, et en particulier en reconnaissance d'actions, les séquences sont de longueur variable dû au fait que les sujets ont des vitesses et des styles de performance différents.

Au début des années 60, Bellman and Kalaba [BK59] ont proposé un algorithme dénommé **DTW** (pour *Dynamic Time Warping*) qui tient compte d'une certaine élasticité lors de la comparaison des séquences, où ces dernières peuvent alors être de longueur variable. L'algorithme DTW a gagné sa popularité en étant extrêmement efficace en tant que mesure de similarité de séries temporelles qui minimise les effets de décalage et de distorsion temporelle en permettant une transformation "élastique" des séries temporelles afin de détecter des formes similaires avec des phases différentes.

En particulier, soit deux séries temporelles X et Y de tailles respectives T_1 et T_2, représentées par les séquences de valeurs $X = \{x_1, x_2, ..., x_{T_1}\}$ et $Y = \{y_1, y_2, ..., y_{T_2}\}$. Comme illustré sur la Figure 2.18, l'alignement de ces deux séquences au moyen du DTW revient à trouver le chemin de coût minimum parmi tous les chemins pouvant aligner X et Y. Ceci pourrait être couteux en temps de calcul en raison de la croissance exponentielle du nombre de chemins optimaux lorsque les longueurs de X et Y croissent linéairement. Pour surmonter ce problème, le DTW utilise l'algorithme basé sur la programmation

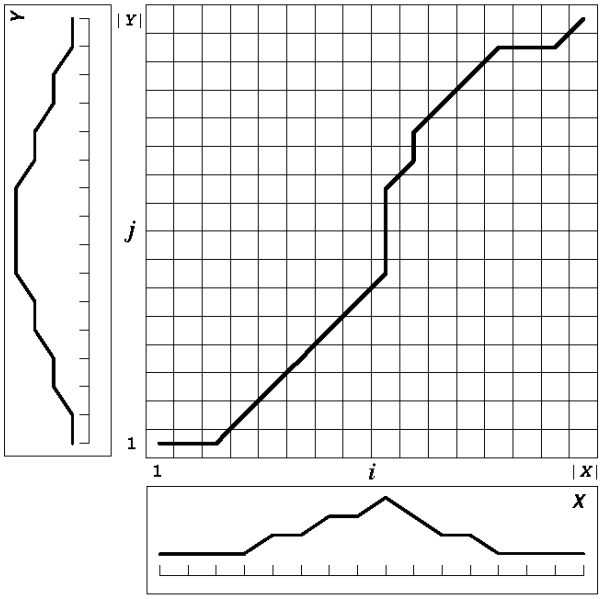

FIGURE 2.18 – Illustration d'une matrice de coût et du chemin de distorsion de distance minimale [cnb18].

dynamique de manière à effectuer cette optimisation suivant une complexité quadratique ($O(T_1T_2)$). L'algorithme commence par construire la matrice de distance $C \in \mathbb{R}^{T_1 \times T_2}$ représentant toutes les distances par paires entre X et Y comme suit :

$$C \in \mathbb{R}^{T_1 \times T_2} : c_{i,j} = \| x_i - y_i \|, i \in [1 : T_1], j \in [1 : T_2] \tag{2.6}$$

Une fois cette matrice obtenue, l'algorithme trouve le chemin d'alignement qui traverse les zones à faible coût dans cette matrice de coûts. Ce chemin d'alignement (ou chemin de déformation) définit la correspondance d'un élément $x_i \in X$ à $y_j \in Y$ en vérifiant la condition aux limites c'est-à-dire assigner les premier et dernier éléments de X et Y l'un à l'autre.

Cet algorithme a été largement utilisé dans différents domaines dont la reconnaissance de la parole [MRR80, SC78], la reconnaissance de l'écriture [IPS+12, EFV07] et la reconnaissance de la langue des signes [KZ07]. Il a été utilisé aussi pour des fins de reconnaissance et d'analyse d'actions sous sa forme originelle [BAM17, MAKD17] ou bien sous une forme adaptée aux cas considérés [DM15, MR06, MAKD18].

Parmi les travaux se basant sur le DTW sous sa forme originelle, il y a l'approche dite **hiérarchique** proposée par Bloom et al. [BAM16, BAM17] pour reconnaître des actions pré-segmentées. Ces auteurs construisent d'abord des modèles d'actions (*action templates*) en ne retenant dans un modèle d'action qu'un nombre réduit de postures dites **postures clés**. Ils sélectionnent aussi, pour chaque type d'action, les parties du squelette participant

le plus à cette action (comme les bras pour une action "coup de poing") et ils calculent enfin une mesure de similarité via le DTW en alignant le mouvement de chaque partie sélectionnée du squelette membre à membre le long respectivement de la séquence d'action de test et des modèles d'actions prédéfinis. Une autre approche, proposée par Morel et al. [MAKD18], opère d'abord un alignement global, c'est-à-dire reposant sur l'ensemble du squelette de la personne, entre deux séquences d'actions. Cette première étape permet d'ajuster temporellement les séquences considérées. Ensuite, des alignements locaux, c'est-à-dire ajustant les positionnements d'une articulation isolée des deux mouvements, permet d'obtenir de nouveaux alignements potentiellement différents du premier. Ces différents alignements permettent ensuite d'extraire une erreur de synchronie entre les membres d'un sujet durant l'exécution de son action. Muller et al. [MR06] proposent d'utiliser le DTW avec des descripteurs géométriques (voir section 2.2.4) pour des fins de classification d'actions squelettiques. Les auteurs affirment que la nature même de ces descripteurs, qui absorbent une partie de la déformation temporelle en ne rapportant que des informations sur les emplacements relatifs entre articulations, sont très adaptés pour être combinés avec le DTW.

Par ailleurs, des travaux proposent d'étendre l'utilisation du DTW de la reconnaissance d'actions pré-segmentées au cas plus complexe de reconnaissance d'actions dans un flot non segmenté. En effet, l'algorithme de DTW dans sa façon originelle ne tient pas compte de la non présence d'action ou au contraire de la présence de plusieurs exemples d'actions dans une séquence de test. A titre d'exemple, une nouvelle version de DTW, dénommée Stream-DTW, est proposée par Dupont et al. [Dup17] pour calculer de façon incrémentale la valeur de similarité entre une séquence de référence et une potentielle action située dans une fenêtre qui glisse sur le flot non segmenté. Comme le relèvent ces auteurs, se baser sur un déploiement classique du DTW impliquerait le recalcul de la valeur de similarité à l'arrivée de chaque nouvelle frame. Ceci induirait alors un temps de calcul important, et empêcherait son utilisation dans un contexte interactif où la reconnaissance doit se faire en temps réel. Comme illustré sur la Figure 2.19, avec le Stream-DTW, les auteurs proposent de rajouter une nouvelle colonne dans la matrice de coût à chaque nouvelle frame en initialisant sa valeur de base à 0. L'idée des auteurs est que l'alignement n'ait pas lieu sur le flot (de test) en entier, mais plutôt sur la fin du flot en choisissant le meilleur début possible. Cette technique permet alors de mettre à jour la valeur de similarité non pas en temps quadratique mais linéaire.

Il est important de mentionner que d'autres métriques de similarité sont utilisées en reconnaissance d'actions notamment pour être plus robustes aux nombreuses valeurs aberrantes (*outliers*) qui résultent de problèmes d'occultation et de mauvaise reconstruction du squelette. En particulier, l'approche proposée par Pazhoumand-Dar et al. [PDLM15]

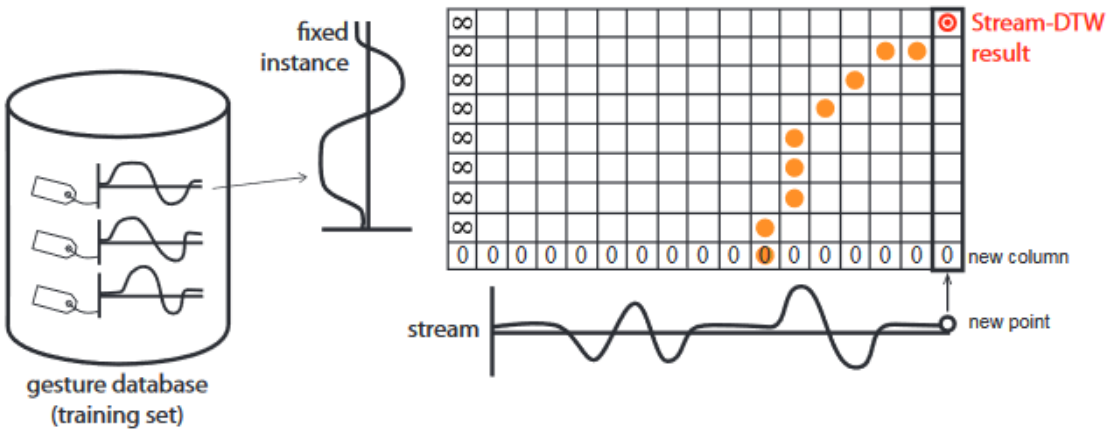

FIGURE 2.19 – Illustration d'une matrice de coût mise à jour de manière incrémentale suivant la technique déployée dans Stream-DTW, proposée dans [Dup17].

extrait d'abord des séquences de positions cartésiennes et d'angles relatifs puis emploie la mesure de similarité dite **LCSS**, dont la tâche est de rechercher la plus longue sous-séquence commune aux séquences à comparer. LCSS est en mesure d'aligner des séquences de différentes longueurs sans pour autant être contrainte d'aligner tous les points. Ainsi, le nombre minimal de points pour lesquels une correspondance est à établir par LCSS est 0 (contrairement à DTW où tous les points doivent correspondre) et le nombre maximal est la longueur de la séquence la plus courte (tous les points de la séquence la plus courte ont des correspondances dans la séquence la plus longue) ce qui permet d'envisager une reconnaissance d'actions dans un flot non segmenté. L'illustration donnée dans la Figure 2.20 permet de relever comment l'alignement au moyen de la LCSS gère les éventuels *outliers* en ne focalisant cet alignement que sur les parties où le mouvement a réellement lieu.

Il est enfin possible de retenir que les approches à base de comparaison élastique partagent plusieurs avantages notamment le fait de nécessiter peu de données pour constituer des modèles d'actions ainsi que la prise en compte d'une certaine élasticité. En revanche, d'une part le DTW ne permet pas de gérer d'éventuels *outliers* retrouvés dans les signaux de par le fait qu'il ne peut ignorer un élément lors de la mise en correspondance des séquences. D'autre part, les mesures de similarité telles que la LCSS, qui ont une plus grande flexibilité pour faire correspondre ou non des points, pourraient conduire à une certaine perte d'information [Mor17]. Ces dernières sont néanmoins plus robustes aux *outliers* que le DTW.

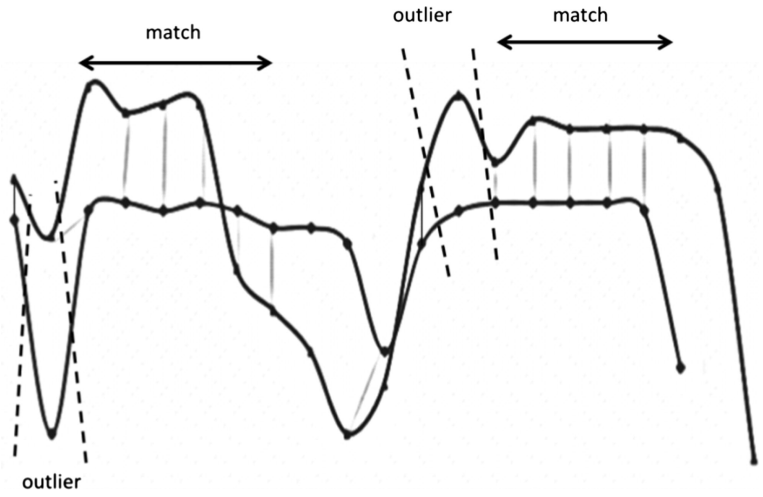

FIGURE 2.20 – Illustration d'une correspondance de deux trajectoires contenant des valeurs aberrantes (*outliers*) au moyen de la LCSS. [PDLM15].

2.3.2 Approches statistiques

Nous nous intéressons dans cette section aux approches proposant de modéliser l'intégralité d'une action sous forme d'un vecteur de descripteurs de taille fixe. A l'opposé des approches séquentielles, les approches statistiques proposent d'intégrer explicitement l'aspect de séquencement temporel dans la représentation d'une action. Nous proposons de décrire ces approches suivant qu'elles conçoivent leur représentation d'action en se basant sur une simple concaténation des données brutes, en extrayant des mesures rapportant des informations haut niveau sur l'action ou bien en procédant par le biais d'un dictionnaire de mots. Avant d'entamer le propre de cette section, en décrivant les différentes approches statistiques, nous proposons d'abord d'introduire les principaux types de classifieurs pouvant être déployés dans une approche statistique. Sur la base des travaux portant sur la reconnaissance d'actions 3D, nous présentons quatre classifieurs très utilisés, à savoir : les machines à vecteurs de support (SVM), le perceptron multicouche (MLP), les k plus proches voisins (k-NN) et les arbres de décision.

Le SVM est un classifieur supervisé destiné à résoudre des problèmes de discrimination et de régression en dressant, comme résultat d'une optimisation quadratique, des limites linéaires ou non linéaires entre plusieurs classes de points (Figure 2.21). Ce classifieur repose sur deux idées clés. La première est la notion de marge maximale. La marge est la distance entre la frontière de séparation et les échantillons les plus proches. Ces derniers sont appelés vecteurs supports. Dans les SVM, la frontière de séparation est choisie comme celle qui maximise la marge. Le problème est de trouver cette frontière séparatrice optimale, à partir d'un ensemble d'apprentissage. Ceci est fait en formulant le problème

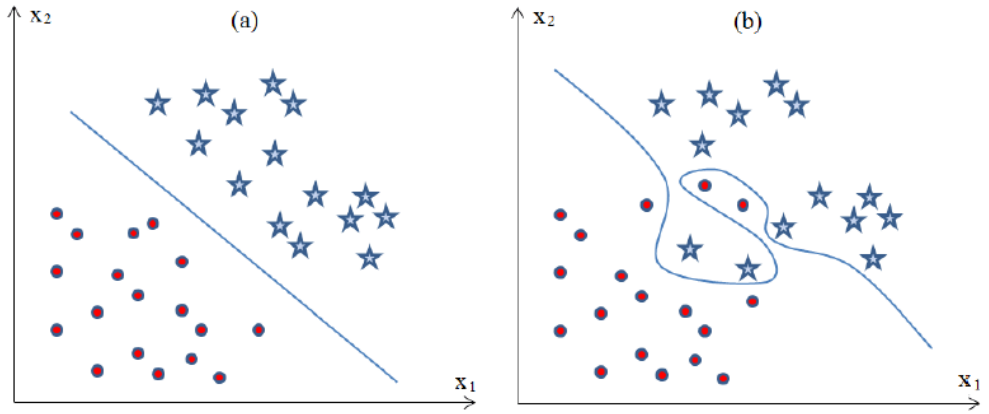

FIGURE 2.21 – Représentation des descripteurs, colorés en rouge ou bleu selon la classe à laquelle ils appartiennent. L'hyperplan de séparation est indiqué par un trait plein, linéaire dans le premier cas et non linéaire dans le second.

comme un problème d'optimisation quadratique. La deuxième idée clé des SVM est de transformer l'espace de représentation des données d'entrée en un espace de plus grande dimension (théoriquement pouvant être de dimension infinie), dans lequel l'existence d'une séparation linéaire est probable. Ceci est notamment intéressant afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables en se basant sur une fonction noyau. Depuis leur introduction dans les années 1990 par Cortes and Vapnik [CV95], les SVM ont rapidement été adoptés en reconnaissance de formes pour leur capacité à modéliser des données de grandes dimensions, pour le faible nombre d'hyperparamètres, leurs garanties théoriques et leurs bons résultats en pratique.

Le perceptron est un modèle inspiré des théories cognitives. Il peut être vu comme le type de réseau de neurones le plus simple. C'est un classifieur linéaire. Ce type de réseau neuronal ne contient aucun cycle (il s'agit d'un réseau de neurones à propagation avant, c'est-à-dire *feedforward*). Dans sa version simplifiée, le perceptron est mono-couche et n'a qu'une seule sortie à laquelle toutes les entrées sont connectées. Le perceptron multicouche (MLP pour *multilayer perceptron*) est un type de réseau neuronal formel organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement. Chaque couche est constituée d'un nombre variable de neurones, les neurones de la dernière couche (dite de sortie) étant les sorties du système global. Il existe d'autres types de modèles à base de neurones très utilisés en reconnaissance d'actions, notamment les réseaux de neurones récurrents, que nous abordons dans la section 2.3.3.

Les k plus proches voisins (k-NN pour *k-nearest neighbors*) est une méthode non paramétrique très utilisée en classification. L'algorithme du k-NN permet d'étiqueter une

instance de classe inconnue par la classe la plus comptabilisée parmi les classes de ses k plus proches voisins. Aucun apprentissage n'est nécessaire pour ce classifieur vu que tous les calculs ont lieu pendant la phase de test, où une nouvelle instance est comparée à toutes les instances de l'ensemble d'apprentissage. Un cas particulier du k-NN consiste à ne tenir compte que d'un seul voisin, donnant lieu au classifieur du plus proche voisin (1-NN). Dans ce cas, la règle consiste à simplement renvoyer l'étiquette de l'instance issue de l'ensemble d'apprentissage et qui permet de minimiser une mesure de distance à définir. L'avantage de cette façon d'opérer est qu'aucun paramètre n'est à définir. L'inconvénient est que cette comparaison peut être coûteuse en temps de calcul où ce dernier est dépendant du nombre d'échantillons disponibles (pouvant être très important). Une manière d'améliorer ce classifieur était de choisir un représentant de chaque classe et de comparer l'instance de test aux représentants des classes et non pas à tous les éléments de l'ensemble d'apprentissage.

L'arbre de décision est un outil d'aide à la décision très utilisé en classification, qui représente un ensemble de choix sous la forme graphique d'un arbre. Les différentes décisions possibles sont situées aux extrémités des branches (les **feuilles** de l'arbre), et sont atteintes en fonction des décisions prises à chaque étape. L'avantage de ce classifieur est qu'il permet de sélectionner automatiquement les variables discriminantes à partir de données non-structurées et potentiellement volumineuses. Ils permettent ainsi d'extraire des règles logiques de cause à effet (des déterminismes) qui n'apparaissaient pas initialement dans les données brutes. Par ailleurs, un modèle de classification plus complexe, appelé forêt aléatoire [Bre01], étend cette procédure pour construire non pas un arbre mais une forêt d'arbres de décision. L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents. Une décision est alors prise en faisant voter l'ensemble des arbres et en choisissant la réponse majoritaire.

2.3.2.1 Représentations brutes

Une manière intuitive de concevoir une représentation statistique d'une action squelettique est de concaténer, dans un seul grand vecteur, les données brutes du squelette issues de chaque frame. Cette manière de procéder peut notamment être justifiée par le fait qu'un squelette comporte à la fois des informations variées (coordonnées cartésiennes 3D, angles articulaires et relations géométriques) et pertinentes (ciblant uniquement le squelette et pas son environnement). Ainsi, de nombreuses représentations ont été proposées suivant ce principe en variant la modalité des données brutes considérées et/ou le nombre de données considérées par frame [YLY14, WZZZ13a, SPSS12, FMKN12, WLW14, YT12, YT14, CK13, OBT13, SSS05].

Ainsi, la représentation dénommée *Eigen-Joints* proposée par Yang and Tian [YT12, YT14] est issue d'une concaténation dans un seul vecteur de plusieurs types de coordonnées cartésiennes relatives (section 2.2.2.2) suivie d'une analyse en composantes principales (ACP). En particulier, l'encodage de la temporalité se résume au calcul de la différence des positions cartésiennes du squelette entre chaque frame et celle qui la précède ainsi que la différence entre chaque frame et la frame initiale de la séquence. Par ce procédé, l'objectif est d'extraire explicitement aussi bien l'information temporelle locale que l'information temporelle globale. De par le grand nombre de combinaisons possibles, le vecteur descripteur pour chaque frame à une dimension de 2970, ce qui représente une dimension très élevée. Les auteurs proposent alors d'appliquer une ACP pour réduire cette dimension en ne prenant que les 128 premières valeurs propres (correspondant aux meilleurs 5% des descripteurs). C'est la raison pour laquelle le vecteur descripteur ainsi formé est appelé *EigenJoints* pouvant être traduit en **articulations propres**. Enfin, un classifieur à base de l'algorithme du plus proche voisin est utilisé pour déterminer l'appartenance de l'échantillon de test, constitué de M frames, comme suit :

$$C^* = \arg\min_{C} \sum_{i=1}^{M} \parallel d_i - NN_C(d_i) \parallel^2 \tag{2.7}$$

où $d_i, i = 1, 2, ..., M$ est le vecteur descripteur contenant les 128 articulations propres, M le nombre de frames et $NN_C(d_i)$ le plus proche voisin de d_i appartenant à la classe C.

Nous relevons qu'à l'image de beaucoup d'autres approches de reconnaissance d'actions, cette représentation souffre d'une grande dimensionalité. Raison pour laquelle les auteurs proposent de réduire la dimension du vecteur descripteur au moyen d'une ACP pour ne retenir que 5% de ses éléments. Or en procédant ainsi, il est fortement probable que la représentation résultante ne tienne plus compte des différents aspects inclus dans la phase initiale, notamment l'aspect temporel. Il aurait été plus intéressant justement de sélectionner quelques paires d'articulations et retenir tous les descripteurs qui en seraient extraits de manière à rapporter effectivement les informations temporelles initialement ciblées.

Un autre exemple d'approches à base de concaténation de données brutes est proposé par Patsadu et al. [PNW12]. Il s'agit simplement de normaliser les coordonnées cartésiennes de vingt articulations de chaque frame et les mettre ensemble dans un seul vecteur. Il est à noter pour cette représentation que l'information temporelle est totalement manquante dans la mesure où l'ordre des descripteurs dans un vecteur n'a pas d'impact sur le résultat de la classification. Les auteurs proposent pour l'étape de classification d'évaluer les performances obtenues séparément par plusieurs classifieurs dont le perceptron multicouche (MLP), les machines à vecteurs de support (SVM) et les arbres

de décision. Les auteurs ont obtenu la meilleure performance au moyen du MLP. Chen et Koskela [CK13] propose de normaliser aussi les positions des articulations de manière à être invariant aux morphologies des sujets mais proposent de concaténer, dans un seul vecteur, non pas les positions absolues mais les positions relatives des articulations. De plus, ce vecteur est augmenté par des descripteurs extraits à partir des images RGB. La classification de la représentation résultante est basée sur une variante du classifieur MLP.

Fothergill et al. [FMKN12] ont proposé une représentation dans le contexte de reconnaissance d'actions dans un flot non segmenté. Leur approche se base sur une fenêtre temporelle glissante de taille égale à 35 frames qui scrute en continu le flot non segmenté. En particulier, pour chaque frame de la fenêtre, un vecteur de 130 dimensions est extrait. Ce vecteur comporte 35 angles, 35 vitesses de rotations et 60 vitesses linéaires. Enfin, pour constituer la représentation finale de la fenêtre en cours, les vecteurs extraits des 35 frames sont concaténés pour former un vecteur global de 4550 dimensions. A chaque instant, ce vecteur de descripteurs est évalué avec un ensemble d'arbres de décisions (nombre d'arbres non spécifié). Dans cette représentation, la temporalité est fortement modélisée, notamment par la présence de vitesses angulaires et linéaires qui représentent plus de 70% de la taille du vecteur de descripteurs. Néanmoins, la taille de représentation finale est très importante et peut impacter négativement les performances de reconnaissance.

La représentation proposée par Sung et al. [SPSS12] est très similaire à la représentation dite *Eigen-Joints* sur le plan de la nature des données qui y sont incluses mais elle diffère par le fait qu'elle résulte d'une concaténation de certaines données squelettiques et pas toutes et qu'aucune sélection a posteriori n'est effectuée. En particulier, la représentation que proposent Sung et al. [SPSS12] est principalement adaptée pour des actions d'interaction avec les mains. Elle comporte des angles relatifs (47 descripteurs par frame), les valeurs maximales et minimales des coordonnées cartésiennes des mains sur les 6 dernières frames de la séquence (16 descripteurs au total) et les angles relatifs entre neuf frames sélectionnées le long de la séquence de manière à être temporellement équidistantes (396 descripteurs au total).

2.3.2.2 Descripteurs haut-niveau

La deuxième catégorie des approches statistiques regroupe les représentations constituées par des descripteurs plus élaborés, calculés à partir des données brutes. Ces descripteurs correspondent à des mesures mathématiques (courbure de trajectoire, moment d'inertie, etc.), faisant ressortir de manière plus explicite les caractéristiques discriminantes des actions.

Dans cette catégorie, il est possible de citer la représentation dite *actionlet* proposée par Wang et al. [WLWY12, WLW14]. L'approche proposée considère pour chaque articu-

lation et à chaque frame, toutes les positions relatives par rapport aux autres articulations. Ceci permet donc de construire une fonction temporelle pour laquelle la transformée de Fourier est appliquée et d'extraire les coefficients de basse fréquence, considérés comme descripteurs. De plus, comme illustré sur la Figure 2.22, cette approche propose d'extraire l'information temporelle en appliquant le processus d'extraction des descripteurs pour l'intégralité de la séquence mais aussi pour des sous-séquences suivant une hiérarchie pyramidale à trois niveaux. Les auteurs entraînent enfin un SVM pour l'étape de classification.

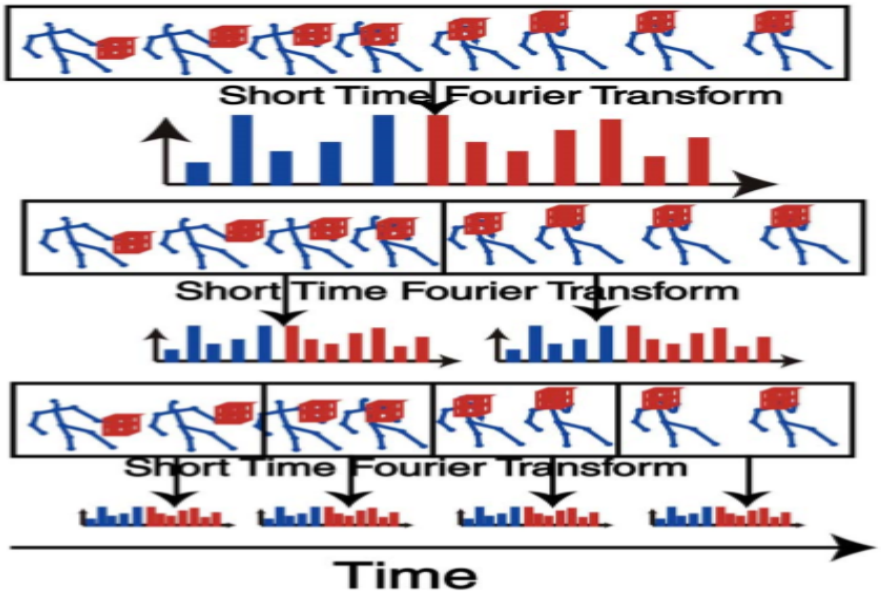

FIGURE 2.22 – Extraction des descripteurs suivant une hiérarchie temporelle à trois niveaux pour former la représentation dite *actionlet* [WLW14].

La représentation proposée comporte au moins trois avantages. D'abord, en éliminant les coefficients de Fourier de haute fréquence, cette représentation est robuste au bruit. Deuxièmement, cette représentation est insensible au problème d'alignement temporel, vu qu'une série temporelle possède toujours les mêmes coefficients de Fourier sur le plan temporel. Enfin, la structure hiérarchique de la procédure d'extraction des descripteurs permet d'encoder efficacement l'information temporelle des actions.

Nous relevons néanmoins que les descripteurs extraits sont, d'une part, tous de même nature (coefficients de Fourier) et, d'autre part, sont extraits pour la totalité des articulations. Il aurait été plus intéressant d'inclure d'autres types de mesures et de réduire le nombre d'articulations considérées.

Une autre représentation dénommée **Cov3DJ** par Hussein et al. [HTGES13] encode l'information temporelle selon la même procédure que celle suivie par Wang et

al. [WLWY12, WLW14] mais en permettant aux sous-séquences d'un même niveau de profondeur de se chevaucher. Cette représentation se base sur le calcul de la matrice de covariance des positions articulaires dans une séquence de frames (Figure 2.23). En particulier, supposons que le corps soit représenté par K articulations, et que l'action soit effectuée sur T frames. Soit $x_i^{(t)}, y_i^{(t)}$ et $z_i^{(t)}$ les coordonnées x, y et z de la i^{eme} articulation et cela à la frame t. Soit S le vecteur de toutes les positions des articulations, c'est-à-dire $S = [x_1, ..., x_K, y_1, ..., y_K, z_1, ..., z_K]'$ comportant en tout $N = 3K$ éléments. Le descripteur de covariance pour la séquence est alors défini comme suit :

$$C(S) = \frac{1}{T-1} \sum_{t=1}^{T} (S^{(t)} - \overline{S})(S^{(t)} - \overline{S})' \tag{2.8}$$

avec $\overline{S}$ la moyenne des $S^{(t)}$ sur toute la séquence S.

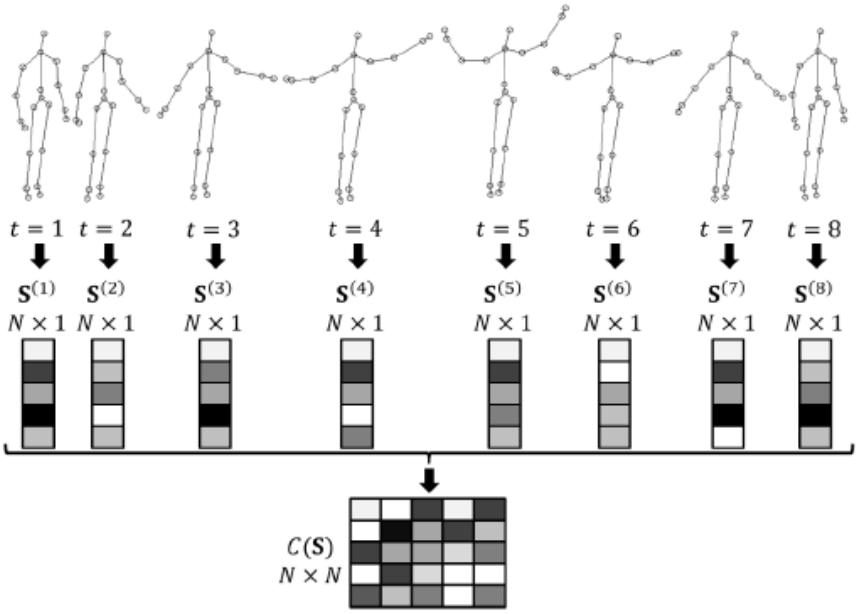

FIGURE 2.23 – Illustration de la matrice de covariance permettant de construire le descripteur **Cov3DJ** [HTGES13].

La matrice de covariance de la séquence S notée $C(S)$ est une matrice $N \times N$ symétrique. Le vecteur descripteur retenu est alors composé des éléments de son triangle supérieur uniquement. Par exemple, pour un squelette de 20 articulations (Figure 2.23), $N = 3 \times 20 = 60$. Le triangle supérieur de la matrice de covariance comporte dans ce cas $N(N+1)/2 = 1830$ éléments, correspondant à la dimension du vecteur descripteur obtenu sur toute la séquence.

Pour encoder l'information temporelle, cette procédure d'extraction est appliquée suivant une hiérarchie temporelle à plusieurs niveaux. A titre illustratif, pour une hiérarchie

à deux niveaux (Figure 2.24), la taille de la représentation finale est de 7320 descripteurs (1*1830 + 3*1830 = 7320). En dépit de cette dimensionalité très importante, qui peut l'être davantage pour une hiérarchie à trois niveaux, les auteurs n'ont effectué aucune réduction. Pour l'étape de classification, les auteurs proposent d'utiliser un SVM.

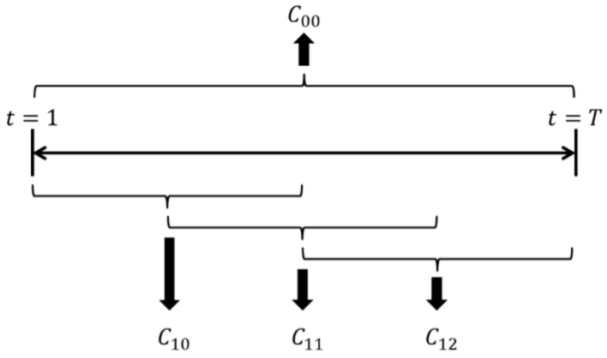

FIGURE 2.24 – Construction temporelle du descripteur de covariance **Cov3DJ**. C_{li} est la i^{eme} matrice de covariance du l^{eme} niveau de la hiérarchie. Une matrice de covariance au l^{eme} niveau couvre $\frac{T}{2^l}$ frames de la séquence, où T est la longueur de la séquence entière [HTGES13].

Gowayyed et al. [GTHES13] proposent de projeter la trajectoire 3D de chaque articulation sur trois plans et de se baser sur un histogramme pour décrire cette trajectoire sur chacun des plans. En particulier, pour chaque trajectoire 2D projetée $S = P_1, P_2, P_3, ..., P_n$, les auteurs proposent de calculer l'angle de direction θ $(t, t+1)$ entre chaque paire de positions successives P_t et P_{t+1} comme suit :

$$\theta(t, t+1) = \arctan \frac{y_{P_{t+1}} - y_{P_t}}{x_{P_{t+1}} - x_{P_t}} \tag{2.9}$$

En se basant sur ces valeurs angulaires, les auteurs proposent de construire un histogramme à K entrées. L'intervalle $[0, 2\pi]$ est subdivisé en K secteurs (bins), de manière à ce que chaque secteur soit relié à une entrée de l'histogramme. Ainsi, à chaque nouvelle valeur de l'angle de direction, l'entrée associée dans l'histogramme est mise à jour. A l'opposé d'un histogramme classique, la mise à jour de cet histogramme ne se fait pas en incrémentant par une unité mais par la longueur du segment reliant les deux points en question, c'est-à-dire P_t et P_{t+1}.

Afin d'incorporer l'information temporelle, les auteurs proposent de construire ces histogrammes hiérarchiquement en divisant à chaque niveau la séquence précedente par deux, de manière à ce que les sous-séquences ne se chevauchent pas (comme proposé par Wang et al. [WLWY12, WLW14]). La représentation finale, dénommée **HOD** (pour *Histogram of Oriented Displacements*), s'obtient par concaténation des trois histogrammes

formés de chaque articulation et pour chaque niveau temporel de la hiérarchie. Ainsi, pour un histogramme ayant $K = 8$ entrées et avec un niveau de profondeur égal à 3, la représentation finale a une dimension égale à 3840 ([3 projections] $\times$ [20 articulations] $\times$ [8 bins] $\times[2^{(3 \quad niveaux)}]$). Les auteurs ont opté pour des SVMs afin d'effectuer la classification.

2.3.2.3 Dictionnaire de mots

Contrairement aux deux catégories d'approches statistiques précédentes, une représentation à base d'un dictionnaire ou sac de mots applique un opérateur de codage pour projeter chaque squelette ou pose en un seul code (ou mot) en utilisant un dictionnaire contenant tous les codes possibles. Cette procédure est également appelée quantification de descripteurs. Pour former la représentation finale, les fréquences des codes sont concaténées dans un seul vecteur de descripteurs. Le codage par sac de mots est très utilisé pour former des représentations squelettiques [LWQ13, JZPQ14, ZLP+13, ZWZ13]. La principale différence entre ces approches est le procédé de création du dictionnaire de mots.

Par exemple, Wang et al. [WWY13] proposent de scinder le squelette en cinq parties (tête, bras gauche, bras droit, jambe gauche et jambe droite) et de construire ensuite un ensemble de dictionnaires de poses (simple ou multiple) pour chacune de ces parties et pour chaque classe d'action. En particulier, l'algorithme k-means est d'abord utilisé pour former les cinq dictionnaires de poses simples pour une classe d'action comme illustré sur la Figure 2.25a. Ensuite, pour mieux extraire l'information spatiale, les auteurs proposent de considérer les classes deux à deux et de construire à partir des dictionnaires de poses simples cinq autres dictionnaires composés de n-uplets de n (2,3,...) poses. Ces n-uplets sont des combinaisons possibles entre des poses figurant dans les dictionnaires simples d'une action, sans nécessité d'ordre temporel précis, mais qui ne figurent pas dans l'autre classe d'action. Ainsi, il est doublement intéressant, d'un point de vue de caractérisation spatiale, de savoir qu'un couple de poses figure dans une classe d'action et pas dans une autre. Plus encore, en suivant la même procédure, cinq autres dictionnaires temporels sont créés en regroupant des n-uplets de poses qui figurent dans une classe d'action dans un ordre temporel précis. Des exemples des éléments composant les dictionnaires spatiaux et temporels sont donnés dans la Figure 2.25b. La représentation finale d'une action est le résultat de la concaténation des projections de cette action sur les dictionnaires spatiaux et temporels des cinq parties du corps. Pour ce qui est de la classification, les auteurs ont entraîné plusieurs SVMs binaires en mode un-contre-un, où la classe finale est celle ayant été le plus prédite par les classifieurs.

Luo et al. [LWQ13] proposent aussi de construire un dictionnaire pour chaque classe mais ils se basent sur un codage parcimonieux. Il s'agit de construire des dictionnaires où

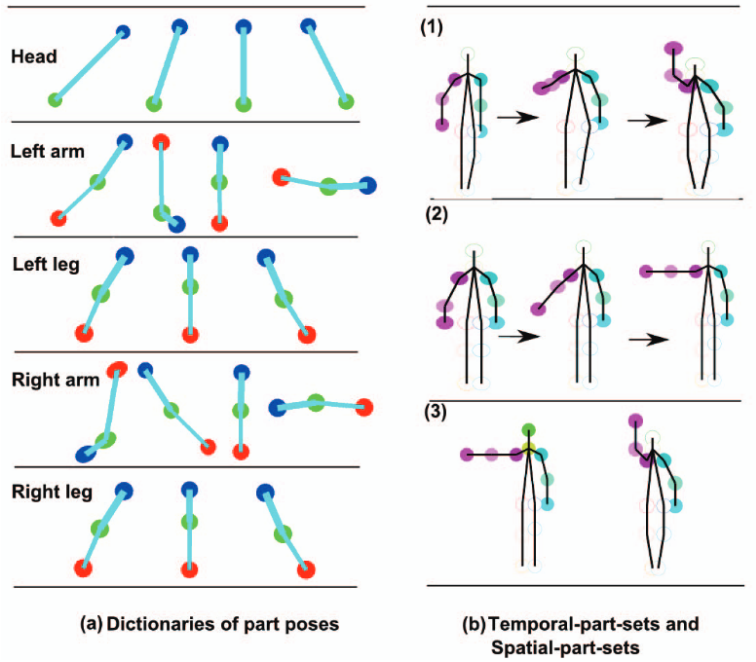

FIGURE 2.25 – Illustration (a) des poses simples, (b)-(1,2) des n-uplets temporels et (c)-(3) des n-uplets spatiaux composant les cinq dictionaires utilisés pour former la représentation de Wang et al. [WWY13].

la décomposition d'une séquence d'action est une combinaison linéaire d'un nombre réduit d'éléments du dictionnaire, appelés atomes. Lors de l'apprentissage des dictionnaires parcimonieux, les auteurs ajoutent une contrainte dite **géométrique** de manière à ce que les atomes du dictionnaire soient géométriquement reliés. Les auteurs affirment que leur représentation, de par sa parcimonie, est en mesure de mieux caractériser une classe d'action. Enfin, pour encoder l'information temporelle, les auteurs se basent sur la technique déjà présentée dite de hiérarchie temporelle. En fait, comme illustré sur la Figure 2.26, les auteurs divisent une séquence en segments non chevauchant à une profondeur égale à 3. Les projections de chacune des sous-séquences ainsi générées sur les dictionnaires de chaque classe d'action sont ensuite concaténées pour former la représentation finale. Un SVM linéaire est ensuite entraîné pour permettre la classification des séquences.

2.3.3 Apprentissage profond

Un autre aspect intéressant permettant de distinguer entre les différentes approches de reconnaissance d'actions est la quantité de données dont ont besoin ces approches pour construire leur modèles. Ainsi, il est possible de distinguer entre les approches nécessitant une faible quantité de données, dont les approches statistiques déjà présentées dans la

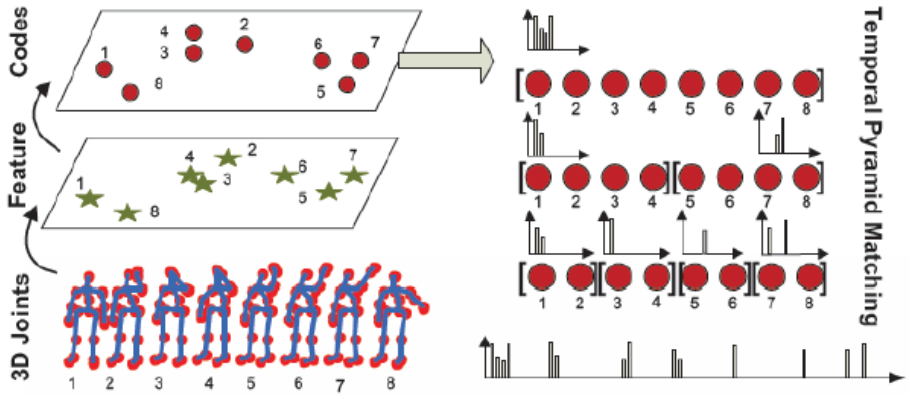

FIGURE 2.26 – Illustration de la construction temporelle pour la représentation à base de dictionnaire proposée dans [LWQ13].

section 2.3.2, et les approches gourmandes en données telles les HMM et plus récemment les réseaux de neurones récurrents en particulier les LSTM. Ayant déjà décrit les approches nécessitant peu de données dans les sections précédentes, nous nous focalisons, dans cette section, uniquement sur les approches à base d'apprentissage profond bien que nous ayons délibérément choisi de ne pas orienter nos travaux vers l'utilisation de ce type d'approches. Ceci est notamment dû au contexte d'application de nos travaux où il n'est pas possible de fournir de grandes quantités de données pour entraîner les modèles. Néanmoins, et au vu des hautes performances que les approches à base d'apprentissage profond permettent de réaliser dans divers domaines dont la reconnaissance d'actions, il nous a semblé nécessaire d'en faire une présentation.

Un réseau de neurones récurrents (RNN) est un réseau de neurones artificiels présentant des connexions récurrentes. Il s'agit en fait d'un réseau constitué d'unités interconnectés, interagissant non-linéairement et pour lequel il existe au moins un cycle dans la structure. Les unités sont reliées par des arcs (synapses) qui possèdent un poids. La sortie d'un neurone est une combinaison non linéaire de ses entrées (Figure 2.27).

Les réseaux de neurones récurrents classiques sont exposés au problème de disparition de gradient qui les empêchent de modifier leur poids en fonction d'événements passés. Lors de l'entraînement, le réseau essaie de minimiser une fonction d'erreur dépendant en particulier de la sortie. Les RNNs se présentent sous différentes variantes. L'architecture la plus utilisée pour répondre au problème de disparition de gradient est le réseau de neurones récurrents à mémoire court-terme et long terme connu sous l'abréviation LSTM. Le réseau LSTM a été proposé par Sepp Hochreiter et Jurgen Schmidhuber en 1997 [SSB14]. L'idée associée au LSTM est que chaque unité computationnelle est liée non seulement à un état caché h mais également à un état c de la cellule qui joue le rôle de mémoire. Le passage

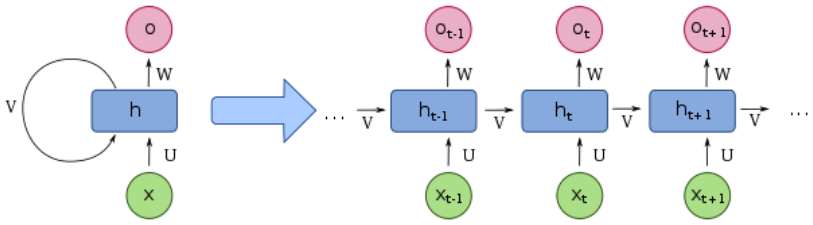

FIGURE 2.27 – Schéma d'un réseau de neurones récurrents à une unité reliant l'entrée et la sortie du réseau. A droite la version dépliée de la structure [Wik18].

de c_{t-1} à c_t se fait par transfert à gain constant et égal à 1. De cette façon, les erreurs se propagent aux pas antérieurs (jusqu'à 1 000 étapes dans le passé) sans phénomène de disparition de gradient.

Une des premières approches de reconnaissance d'actions squelettiques à base d'apprentissage profond est celle proposée par Du et al. [DWW15]. Comme illustré sur la Figure 2.28, il s'agit d'une approche à plusieurs couches. Les auteurs proposent d'abord de subdiviser le squelette en cinq parties se rapportant aux deux bras, aux deux jambes et au torse. Dans la première couche, les données de chacune de ces parties servent à entraîner un réseau de neurones récurrents bidirectionnel (BRNN). Afin de modéliser les parties voisines du squelette, par exemple le bras gauche et le tronc, ils combinent dans la deuxième couche la représentation issue du BRNN relatif au tronc avec celles des quatre autres BRNN pour obtenir quatre nouvelles représentations. De manière similaire à la première couche, les quatre représentations résultantes sont fournies séparément à quatre réseaux BRNN dans la troisième couche. Cette procédure de fusion et d'apprentissage est

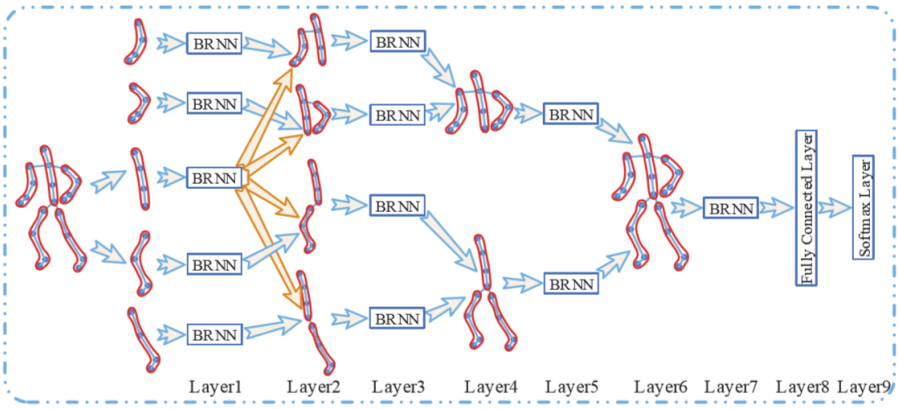

FIGURE 2.28 – Un diagramme illustratif du réseau hiérarchique de neurones récurrents proposé dans [DWW15].

réitérée pour les couches quatre, cinq, six et sept, pour obtenir la représentation finale du squelette reconstitué. Du point de vue de l'apprentissage des caractéristiques, ces BRNN empilés peuvent être considérés comme extracteur des descripteurs spatiaux et temporels de la séquence d'action. Après avoir obtenu ces descripteurs, une couche complètement connectée et une couche Softmax sont rajoutées pour classifier l'action.

Zhu et al. [ZLX+16] ont poussé plus loin l'idée de Du et al. [DWW15] en proposant une architecture qui détermine automatiquement les assemblages les plus pertinents d'articulations au lieu de spécifier à l'avance comment les articulations doivent être groupées. En particulier, l'architecture illustrée dans la Figure 2.29 se compose de trois couches de LSTM, de deux couches de fusion des représentations intermédiaires et d'une dernière couche de Softmax pour la classification.

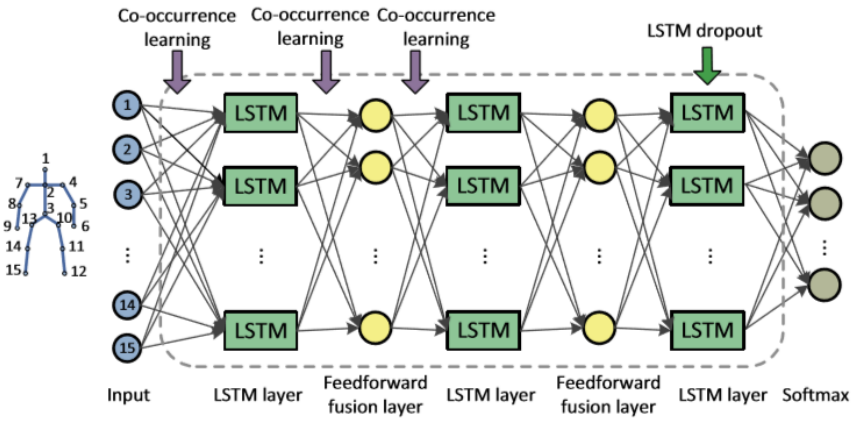

FIGURE 2.29 – Illustration de l'architecture proposée par [ZLX+16] permettant la recherche automatique des relations de co-occurence entre les articulations.

Plus récemment, Liu et al. [LSX+17] proposent une amélioration des LSTM de manière à mieux encoder l'information spatiale caractérisant un squelette. En effet, les LSTM ont été développés pour principalement modéliser la progression temporelle d'une série. Or, une action squelettique comporte aussi un aspect spatial et donc la disposition des articulations en entrée d'une architecture ne devrait pas être aléatoire. Pour remédier à cela, les auteurs proposent de modifier la structure interne d'un nœud LSTM de manière à ce que l'optimisation des poids lors de l'entraînement s'effectue sur le plan temporel (comme c'est déjà le cas) mais aussi spatial suivant une traversée du squelette permettant de tenir compte de la structure de ce squelette. Plus encore, les auteurs proposent d'enrichir chaque noeud du LSTM avec une porte dite **porte d'authentification** (*trust gate*) qui mesure la fiabilité d'une donnée en entrée à chaque étape d'optimisation spatio-temporelle.

Enfin, il convient de mentionner que toutes les approches à base d'apprentissage profond ne suivent pas toutes une approche séquentielle. En effet, certains travaux sont plutôt

des approches statistiques et se basent plutôt sur des réseaux de convolution. En effet, les réseaux de convolution ont permis d'atteindre de très hauts scores pour des tâches telles que la classification d'images et représentent de puissants extracteurs de descripteurs spatiaux. Motivés par ce potentiel, Li et al. [LZXP17] proposent de transformer une séquence d'action en une image RGB à partir de laquelle des réseaux de convolution extraient des descripteurs. L'architecture proposée comporte une dernière couche de réseau complètement connecté et un Softmax pour la classification. Comme illustré sur la Figure 2.30, d'une part l'architecture permet d'extraire des descripteurs supposés encoder l'information spatiale en utilisant notamment les coordonnées cartésiennes absolues pour former l'image RGB. D'autre part, l'information temporelle est extraite en utilisant les coordonnées relatives entre des squelettes de frames successives pour former l'image RGB. Une fois extraits, ces deux sous-ensembles de descripteurs sont concaténés pour former un seul grand vecteur, utilisé lors de la classification.

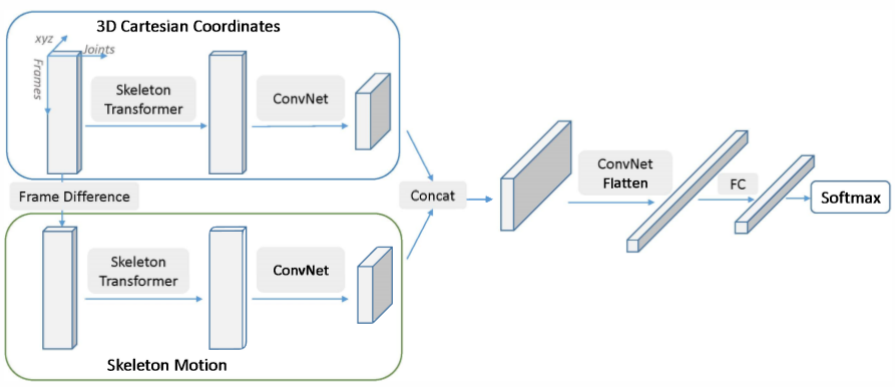

FIGURE 2.30 – Représentation CNN des séquences du squelette pour la classification des actions comme proposée dans [LZXP17].

2.3.4 Discussion

Nous nous sommes intéressés dans cette section au deuxième maillon du processus de reconnaissance d'actions. Il s'agit de décrire comment une action consistant en une séquence de frames peut être modélisée et reconnue. Nous avons notamment fait le distinguo entre d'une part, les approches séquentielles, qui tiennent compte implicitement de la nature séquentielle d'une d'action et d'autre part, les approches statistiques qui extraient un vecteur de descripteurs de taille fixe en intégrant explicitement la dimension temporelle. Nous avons séparément présenté des approches à base d'apprentissage profond qui ont la particularité de nécessiter une quantité importante de données d'apprentissage.

Pour illustrer la catégorie des approches séquentielles, nous avons présenté des approches à base de Modèles de Markov Cachés (HMM) ainsi que des approches à base de comparaison élastique. Dans un second temps, nous nous sommes penchés sur les approches statistiques en les catégorisant dans trois sous-familles suivant qu'elles conçoivent leur représentation d'action en se basant sur une simple concaténation des données brutes, en extrayant des mesures rapportant des informations haut niveau sur l'action ou bien en procédant par le biais d'un dictionnaire de mots. Enfin, nous avons présenté des approches à base d'apprentissage profond aussi bien celles de nature séquentielle (réseaux récurrents) que statistique (réseaux de convolution).

Les représentations que nous avons développées au cours de nos travaux s'inscrivent de manière globale dans la catégorie des approches statistiques et en particulier dans la sous-famille des approches extrayant des mesures haut-niveau. Ce choix est motivé par plusieurs raisons reliées à notre domaine d'application qu'est l'interaction Homme-Machine. Ainsi, une première raison concerne la faible quantité de données disponibles dans ce type d'application. En effet, les modèles statistiques, notamment ceux à base de SVM, sont réputés pour leur capacité de généralisation à partir de très peu de données, alors que dans le même temps les modèles séquentiels comme les HMM présentent des risques de sur-apprentissage dans le cas où il y a peu de données. Une autre raison est relative à notre volonté d'évaluer le potentiel de descripteurs 2D (à extraire à partir de trajectoires 2D) pour modéliser des trajectoires 3D. Cette volonté est d'une part justifiée par l'abondance de jeux de descripteurs 2D très performants et d'autre part par l'intérêt scientifique que peut avoir un transfert du savoir-faire de la 2D vers la 3D.

2.4 Détection d'actions squelettiques 3D non segmentées

Nous nous intéressons dans cette section au quatrième et dernier aspect permettant de distinguer les approches de reconnaissance d'actions. Il s'agit en fait de la reconnaissance d'actions dans un flot non segmenté dite **détection d'actions**. Nous présentons dans cette section les principales approches ayant porté sur cette problématique.

Bien que la détection d'actions dans un flot continu ait un champ d'application plus vaste que la simple reconnaissance d'actions pré-segmentées, la plupart des travaux utilisant des données squelettiques ont plus porté sur la reconnaissance des actions et beaucoup moins sur la détection d'actions. Ceci est notamment dû au fait que la détection est plus complexe, car le système doit assurer deux tâches, à savoir la segmentation et la reconnaissance. Dans ce qui suit, le terme de détection fera référence à la reconnaissance d'actions

dans un flot non segmenté.

Les approches de détection d'actions retrouvées dans la littérature combinent différemment les tâches de segmentation et de reconnaissance. Tout d'abord, il est possible de distinguer les approches **hors ligne** et les approches **en-ligne**. La principale différence entre ces deux familles d'approches est que pour les approches hors ligne la segmentation et la reconnaissance sont effectuées séparément. C'est souvent le cas pour des applications de vidéosurveillance où le flot est analysé après enregistrement à la recherche d'un événement particulier comme une intrusion. Au contraire, une approche de détection en-ligne doit combiner les opérations de segmentation et de reconnaissance car le traitement s'effectue en temps réel. Ceci est notamment indispensable dans un contexte d'interaction Homme-Machine, comme celui dans lequel nous nous situons. Nous nous focalisons ainsi dans cette section sur les approches de détection d'actions en-ligne (**OAD** pour *Online Action Detection*).

En outre, la distinction entre les approches OAD peut être faite sur la manière de combiner la segmentation et la reconnaissance. En effet, il y a les approches où la segmentation s'opère de manière implicite alors qu'il existe d'autres approches qui explicitement segmentent le flot d'entrée. Les approches de segmentation implicite se basent souvent sur des modèles séquentiels tels que les HMM ou les LSTM alors que les approches de segmentation explicite opèrent au moyen de **fenêtre temporelle glissante**. Par exemple, Li et al. [LLX+16] ont proposé une architecture de bout en bout de réseaux de neurones récurrents (RNN) avec une fonction objectif de classification et de régression conjointe pour localiser avec précision les instants de début et de fin des actions. Bien que les performances affichées soient intéressantes, ce type d'architectures nécessite de grandes quantités de données d'apprentissage, qui ne sont pas toujours disponibles. Comme évoqué dans les sections précédentes, nos travaux ne se basent pas sur des modèles séquentiels mais plutôt sur des modèles statistiques et donc la segmentation est opérée de manière explicite. Nous nous focalisons donc sur les approches OAD à base de segmentation explicite.

Pour répertorier et décrire les approches OAD opérant une segmentation explicite, il est possible de se baser sur deux critères. Le premier critère est relatif à la nécessité ou non d'avoir une pose de référence qui sépare deux instances d'actions successives. Le second porte sur la procédure de définition du nombre et la taille des fenêtres temporelles glissantes qu'emploient ces approches. Les approches OAD à segmentation explicite sont décrites ci-après suivant ces deux critères.

2.4.1 Recherche de postures de référence

Le passage d'une problématique de reconnaissance d'actions pré-segmentées à celle de détection (segmentation et reconnaissance) d'actions dans un flot non segmenté est caractérisée par la non connaissance des instants de début et de fin des différentes actions qui peuvent être effectuées.

Une technique consiste à définir une posture représentant une position de repos à laquelle le sujet doit revenir à la fin de la performance d'une action (Figure 2.31). Ainsi, la détection de cette posture marque la fin de l'action et permet de n'enclencher la reconnaissance que sur les frames précédentes. Par exemple, Huang et al. [HYWDLT14] proposent de rajouter une classe "Repos" à l'ensemble des classes à reconnaître de manière à ne lancer la reconnaissance que lorsque la classe "Repos" n'est pas détectée. Néanmoins, cette contrainte peut vite devenir encombrante dans un contexte d'interaction Homme-Machine (un jeu interactif par exemple) où le sujet peut enchaîner plusieurs actions sans s'arrêter.

D'autres approches proposent au contraire de mesurer le degré de certitude de la décision émise à chaque instant pour savoir si une action est, oui ou non, en train d'être effectuée. Par exemple, Zhao et al. [ZLP+13, ZLP+14] proposent d'extraire des descripteurs suivant une fenêtre temporelle glissante et d'entraîner un SVM de façon à ce qu'à chaque instant, le score de la classe potentielle prédite est comparé à un seuil. Si le score est supérieur à ce seuil, alors la frame appartient à la classe prédite, sinon aucune décision n'est émise.

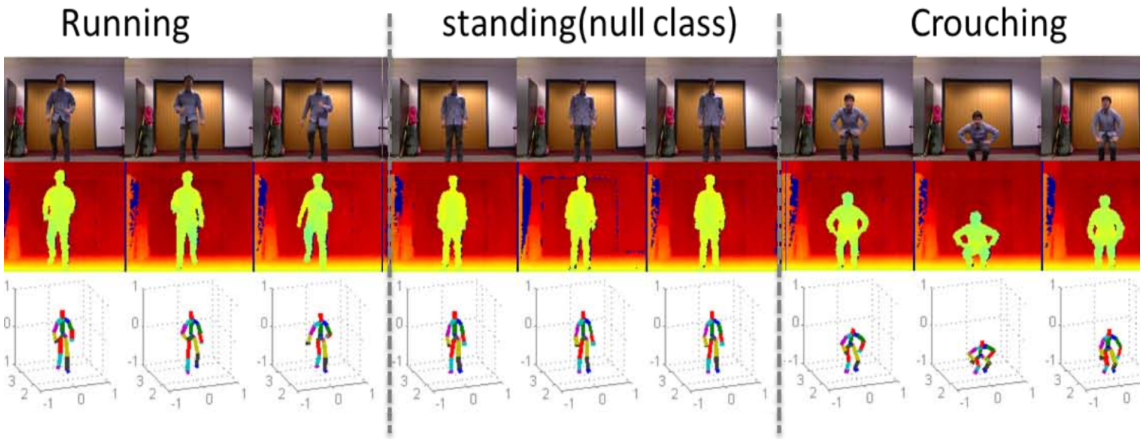

FIGURE 2.31 – Des exemples de frames de la base de données MAD (Multi-Modal Action Detection) illustrant un enchaînement typique (Action-Repos-Action) dans les séquences de cette base [HYWDLT14].

2.4.2 Utilisation de fenêtres glissantes

Parmi les approches OAD basées sur des fenêtres glissantes, nous pouvons distinguer trois familles d'approches.

Les premières méthodes OAD utilisaient des fenêtres glissantes temporelles de taille fixe. Dans [FMKN12], une fenêtre glissante temporelle de 35 frames a été utilisée avec une forêt d'arbres de decision comme classifieur. Zhao et al. [ZLP+13, ZLP+14] ont employé aussi une fenêtre temporelle glissante de 35 frames suivant laquelle une nouvelle représentation a été extraite, appelé SSS (*Structured Streaming Skeleton*).

Zanfir et al. [ZLS13] ont proposé une autre représentation basée sur le descripteur de **pose mobile**. Ce dernier permet de capturer à la fois les positions des articulations et leurs vitesses de déplacement sur de petites fenêtres temporelles de taille fixe. Une autre méthode proposée par Devanne et al. [DBP+17, DWP+15] subdivise en un premier temps le mouvement situé dans chaque fenêtre temporelle en mouvements élémentaires. Ces mouvements élémentaires, appelés **segments de mouvements** (MS), sont appris au préalable et peuvent être considérés comme les éléments d'une base de projection. Dans un second temps, les MS extraits sont modélisés via un classifieur bayesien naïf pour identifier une action en cours, si elle a lieu. Cependant, en raison de la variabilité du style d'exécution, il n'est pas possible de trouver une taille générique de fenêtres temporelles.

De ce fait, la deuxième famille d'approches consiste à considérer une fenêtre de taille croissante en combinant une fenêtre temporelle initiale de taille minimale avec les nouvelles frames qui arrivent. Cela a été proposé par Huang et al. [HYWDLT14] sous forme d'un ensemble de détecteurs d'événements séquentiels appelés **SMMED**. Cette méthode consiste à rejeter séquentiellement les classes les plus improbables en analysant les segments de taille croissante (Figure 2.32). Ce processus s'arrête au moment où il ne reste plus qu'une seule classe. Cette classe est en fait la décision finale de détection. Une approche similaire basée sur le modèle bayésien naïf multinomial a été introduite dans [EMS16]. Bien qu'étant innovantes, ces approches comportent un problème principal relatif à la détermination de la taille minimale de la fenêtre. En effet, si la taille est trop grande, certaines classes ne seront jamais prédites. Si, au contraire, la taille est trop petite, le système risque d'être trop souvent réinitialisé alors même qu'il n'a pas assez d'information pour décider.

Une troisième famille d'approches regroupe les méthodes utilisant plusieurs fenêtres. A titre d'exemple Sharaf et al. [STHES15] ont proposé de mettre en compétition trois fenêtres temporelles glissantes de tailles différentes afin d'accroître les chances de détecter des actions d'une même classe effectuées à différentes échelles temporelles (Figure 2.33). Les auteurs ont alors analysé les données d'apprentissage afin de sélectionner trois tailles

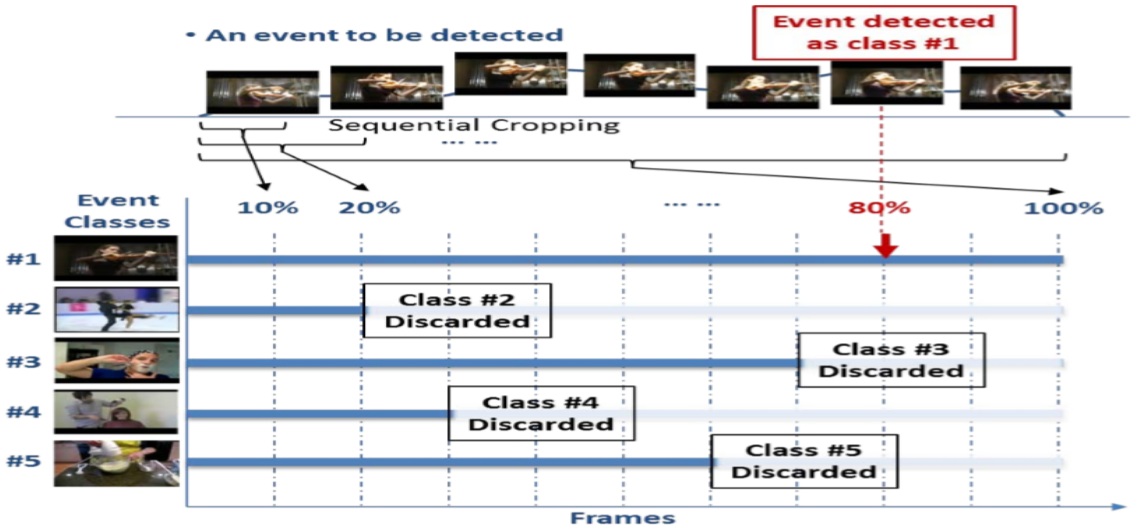

FIGURE 2.32 – Étant donnée une instance de test (séquence d'un sujet jouant du violon en haut de la figure), SMMED rejette automatiquement une classe quand elle est sûre que l'action en cours ne peut pas appartenir à cette classe [HYWDLT14].

en termes de nombre de frames : minimale, moyenne et maximale. Des descripteurs sont ensuite extraits suivant chacune de ces tailles et sont passés à des SVMs.

Une approche similaire à plusieurs niveaux, mais plus efficace sur le plan du temps

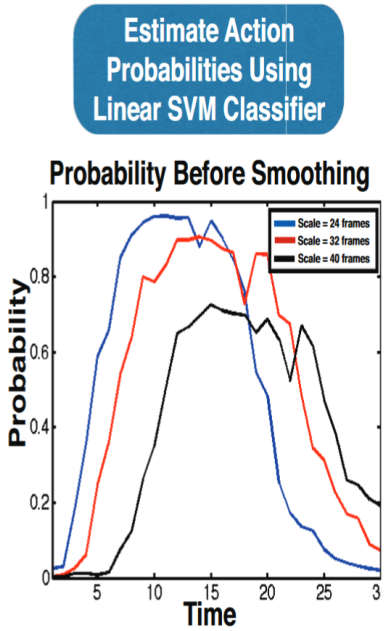

FIGURE 2.33 – Illustration des scores de confiance obtenus par trois classifieurs SVM à différentes échelles temporelles, à savoir 24, 32 et 40 frames [STHES15].

de calcul, a été présentée dans [MHT15, MHT16]. Les auteurs ont proposé d'identifier à chaque position (frame) le sous-intervalle se terminant à la frame courante ayant la plus grande somme. Si cette somme dépasse un seuil donné, le système décide qu'une action est en cours d'exécution. Cependant, on peut se demander pourquoi n'utiliser que trois échelles temporelles alors qu'il devrait y avoir presque autant d'échelles temporelles que de classes d'actions voire même que d'échantillons vu que les styles d'exécution peuvent être très différents.

De plus, en utilisant des fenêtres temporelles de taille fixe ou croissante, ces approches supposent implicitement que la durée des actions pourrait être connue à l'avance. De telles solutions ne sont donc que partiellement satisfaisantes, étant donné que les actions à détecter dans des cas d'utilisation réelle diffèrent grandement en termes de nature du mouvement et de quantité de déplacement. Autrement dit le temps d'exécution peut différer d'une classe à une autre mais aussi d'un sujet à un autre pour une même classe.

2.5 Conclusion

La reconnaissance d'actions 3D est une problématique importante dans le domaine de la vision par ordinateur et la reconnaissance de formes. Comme présenté dans ce chapitre, il existe un grand nombre de travaux proposant d'adresser cette problématique selon différents points de vues et pour des fins diverses. Nous nous sommes en particulier intéressé à trois types de tâches : la reconnaissance d'actions pré-segmentées, la détection en-ligne d'actions non segmentées et la détection précoce d'actions.

La plupart des travaux existants se sont focalisés sur la tâche de reconnaissance d'actions 3D pré-segmentées. Il s'agit d'affecter un label à une observation sensorielle rapportant une action d'un agent humain. Cette tâche est en fait une opération de classification vu que le début et la fin sont connus à l'avance et que le traitement est hors ligne.

Il existe néanmoins certains travaux qui proposent d'adresser le problème plus complexe de détection en-ligne d'actions non segmentées. Contrairement à la reconnaissance d'actions, durant laquelle on classifie des actions pré-segmentées, la détection d'actions est une tâche opérée sur des séquences non segmentées. Le but est de détecter si une action donnée se produit, et si c'est le cas, d'en déterminer la classe (reconnaissance) ainsi que le début et la fin (segmentation). Une contrainte supplémentaire qui complexifie cette tâche est que les traitements doivent être effectués en temps réel (en-ligne) en simulant l'arrivée des frames comme dans un cas d'utilisation réelle.

Enfin, quelques références très récentes proposent de s'intéresser au problème encore plus complexe de détection précoce d'actions non-segmentées. Il s'agit en fait de l'objectif ultime de la détection en-ligne d'actions. Durant cette opération, une décision est attendue le plus tôt possible, idéalement au tout début de l'action, afin de permettre une interaction temps réel.

Dans ce chapitre, nous avons présenté les travaux relatifs à la reconnaissance d'actions squelettiques selon quatre aspects distincts du processus de reconnaissance. Ainsi, le premier aspect porte sur le type de données utilisées en entrée : les coordonnées cartésiennes (absolues ou relatives), les coordonnées angulaires (absolues ou relatives), les relations géométriques ou plusieurs modalités. Le deuxième aspect est relatif à la manière d'appréhender la nature séquentielle des données gestuelles. Le troisième aspect est la quantité de données nécessaire pour entraîner le modèle en question. Le dernier aspect concerne la distinction entre les approches de reconnaissance d'actions dans un flot pré-segmenté et les approches de détection permettant la reconnaissance d'actions dans un flot continu (non segmenté).

En adoptant ces quatre critères, il est possible de situer les travaux présentés dans cette thèse comme suit :

— Nous utilisons les **coordonnées cartésiennes squelettiques** des articulations. Ces coordonnées permettent de constituer une trajectoire sur laquelle nous nous basons pour caractériser une action.
— La représentation que nous concevons encode le séquencement de manière **explicite**. Ainsi notre représentation est un vecteur de même taille quelle que soit la longueur de l'action considérée.
— Notre approche est **transparente** et ne nécessite que très peu de données d'apprentissage. Cette approche est notamment adaptée pour un contexte d'interaction Homme-Machine où un utilisateur peut facilement personnaliser une interface avec une nouvelle commande (nouvelle classe) en n'ayant à fournir que très peu d'exemples et ne nécessitant que peu de temps d'apprentissage.
— Comme énoncé en introduction, notre objectif est de concevoir une approche opérant sur un flot non segmenté, c'est-à-dire une approche de détection. Néanmoins, développer une approche de reconnaissance est souvent une étape intermédiaire avant de proposer une approche de détection. Ainsi, nous présentons dans cette thèse des approches de **reconnaissance d'actions pré-segmentées** et des approches de **détection d'actions dans un flot non segmenté**.

En outre, en analysant les travaux de l'état de l'art, nous avons pu relever plusieurs points intéressants :

— Les approches de la littérature cherchant à modéliser des actions 3D et à les reconnaître ne traitent pas toutes de la même problématique. Il convient donc de déterminer au préalable la finalité visée afin d'identifier les difficultés associées et y répondre adéquatement ;
— Alors que le choix du modèle de classification sur lequel se base une approche donnée est important, il est encore plus important d'avoir une représentation à même de modéliser une action et d'en extraire les informations les plus caractéristiques ;
— La problématique de reconnaissance d'actions fait partie d'un corpus plus large qu'est la reconnaissance de patterns séquentiels. On peut y trouver d'autres problématiques qui présentent des similitudes avec celle que nous proposons d'adresser. Il peut donc être intéressant de tirer profit d'avancées réalisées dans d'autres domaines de la modélisation et la reconnaissance de patterns, autre que celui de la reconnaissance d'actions ;
— De récentes approches, comme celles inspirées de représentations dites biologiques [COK+13, ZP15], essayent progressivement d'inclure l'aspect physiologique de ce problème particulier, en évitant par conséquent de le considérer comme un problème relevant uniquement du domaine de la reconnaissance des formes ;

Sur la base des constatations relevées, nous avons défini le cadre global dans lequel doivent s'inscrire nos travaux de recherche. Les différents éléments le constituant sont présentés ci-après :

- Pour cette étude, nous avons retenu trois tâches relatives à la problématique de reconnaissance d'actions 3D : la reconnaissance d'actions pré-segmentées, la détection en-ligne d'actions non segmentées et la détection précoce d'actions.
- Pour la première problématique portant sur la reconnaissance d'actions 3D pré-segmentées (chapitre 3), nous nous focalisons davantage sur la conception d'une nouvelle représentation d'actions 3D plutôt que sur le moteur de classification. En effet, avoir une bonne représentation constitue un pré-requis important sur lequel nous nous sommes concentrés.
- Pour la deuxième et troisième problématiques traitant de la détection en-ligne et de la détection précoce, nous nous sommes focalisés sur comment adresser les problèmes de variabilités spatio-temporelles.
- Dans cette étude, nous allons identifier les problèmes auxquels nous nous intéressons et nous présenterons des approches nouvelles et "transparentes" en répondant d'une manière explicite à chacun des problèmes retenus.
- Enfin, nous avons relevé que la reconnaissance de tracés manuscrits représente un domaine de recherche où l'on modélise des trajectoires voisines à celles des actions 3D squelettiques. En effet, nous avons noté que les actions 3D à base de squelette partagent plusieurs propriétés avec les tracés manuscrits car tous deux résultent d'une performance humaine. Par exemple, dans les deux cas, il faut modéliser des trajectoires des parties du corps et gérer des problèmes de variabilité. Or, la problématique de reconnaissance de tracés manuscrits a été traitée avant la reconnaissance d'actions 3D, et de ce fait il existe aujourd'hui un grand nombre d'approches qui peuvent être source d'inspiration. Ceci suggère que les difficultés rencontrées pour les actions 3D pourraient être abordées en transposant des solutions déjà proposées pour la reconnaissance de tracés manuscrits. A notre connaissance, ce parallèle entre actions 3D et tracés 2D n'a pas été étudié explicitement. C'est une des contributions de ce travail de thèse.

Chapitre 3

Reconnaissance d'actions 3D pré-segmentées

3.1 Introduction

La reconnaissance d'actions 3D est un sujet de recherche très actif dans le domaine de la vision par ordinateur et la reconnaissance de formes. Ce sujet concerne en fait une large gamme d'applications potentielles telles que la surveillance, l'analyse de mouvements sportifs, l'interaction homme-machine, les jeux, etc. En dépit du grand nombre de travaux menés en ce sens et les nombreux progrès déjà réalisés, cette problématique est loin d'être épuisée.

Techniquement, une action est une séquence générée par un agent humain lors de l'exécution d'une tâche. La reconnaissance d'actions consiste à étiqueter une séquence de mouvement en se basant sur les classes d'actions de référence disponibles. L'expérience souvent citée de Johansson [Joh73] a montré que les humains peuvent reconnaître des actions en observant seulement les articulations principales d'un corps humain (Figure 3.1). Cette observation a motivé l'émergence de nombreuses approches à base de données squelettiques. Les premières approches squelettiques utilisaient en entrée des images 2D, capturées par une caméra RGB, à partir desquelles il était nécessaire d'extraire la structure du squelette. Cependant, une telle extraction 3D à partir de captures vidéo 2D était difficile car les données RGB sont hautement sensibles à divers facteurs tels que les changements d'illumination, les variations du point de vue, les occultations, etc. De nombreux chercheurs ont alors utilisé des systèmes de capture de mouvement pour extraire les positions articulaires 3D. Ces systèmes sont composés de marqueurs et de plusieurs caméras de haute précision. Ils fournissent des mesures précises qui permettent de calculer les centres articulaires. Le travail de traitement de ces données peut être souvent fastidieux

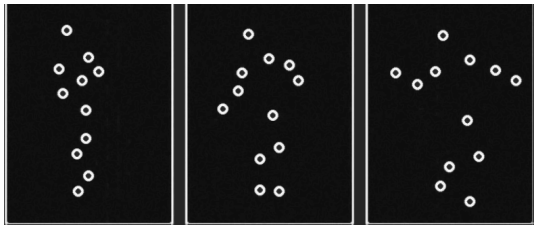

FIGURE 3.1 – Trois frames statiques extraites d'une séquence de marche d'un humain [Joh73].

et ce type de système est très coûteux.

Récemment, l'émergence de caméras de type Kinect [Zha12], qui se basent sur des images de profondeurs, et l'algorithme de reconstruction de Shotton et al. [SSK+13] ont grandement facilité l'extraction des positions articulaires 3D. Cette avancée a suscité un regain d'intérêt pour la reconnaissance d'actions humaines squelettiques. Depuis, il y a eu plusieurs travaux qui proposent de nouvelles représentations pour modéliser une action à partir de positions squelettiques (section 2.2).

Pour l'ensemble des travaux effectués dans le cadre de cette thèse, nous tenons à préciser deux points importants :

— Nous nous intéressons à la modélisation des actions dans un large panel d'applications indépendamment du nombre de trajectoires induites. Il peut s'agir donc des actions effectuées par les bras et les jambes, les bras seuls, les jambes seules, la main seule, etc. Ceci vient en opposition aux approches qui peuvent être uniquement appliquées aux actions impliquant le même nombre de trajectoires.
— Nous nous basons uniquement sur les données squelettiques pour décrire les actions d'un sujet. Les autres modalités, telles les données RGB ou les images de profondeurs, n'ont pas été utilisées dans notre étude. Ces modalités pourraient être envisagées dans des travaux futurs.

Dans ce chapitre, nous présentons deux nouvelles représentations d'actions 3D présegmentées utilisant des données squelettiques. La principale idée supportant ces nouvelles représentations est le transfert du savoir-faire des approches de modélisation de tracés manuscrits à la modélisation d'actions 3D. En effet, nous avons observé que les actions squelettiques partagent plusieurs propriétés avec des tracés manuscrits puisque tous deux résultent d'une performance humaine qui peut, dans les deux cas, être modélisée par des trajectoires d'une ou plusieurs articulations, sensibles à la variabilité de styles. Cette observation suggère donc que la résolution des difficultés liées à la modélisation d'une action squelettique peut s'inspirer de solutions déjà proposées pour la reconnaissance de tracés manuscrits. Nous présentons dans les sections qui suivent cette démarche.

3.2 Transfert de la problématique de reconnaissance d'actions 3D à l'espace des motifs manuscrits 2D

Une première manière simple de s'ancrer sur le savoir-faire de la reconnaissance de tracés manuscrits consiste à ramener le problème de modélisation d'actions 3D dans un espace de représentation 2D. Ceci permet notamment d'appliquer directement les techniques de modélisation de tracés manuscrits.

Avant d'étudier plus en détails cette première approche "naïve", nous proposons d'abord de discuter certaines des difficultés relatives à la représentation d'actions 3D. Ceci permet de préciser les aspects que nous allons considérer lors de la conception de notre représentation, en attirant l'attention sur la complexité de la tâche de reconnaissance d'actions 3D. Il est à noter que la formulation explicite de ces difficultés représente déjà une contribution. Nous pensons qu'il faut absolument expliciter les questions sous-tendues par ce défi pour bien le circonscrire. En effet, les travaux de l'état de l'art omettent souvent d'expliciter ce qui rend difficile la tâche de reconnaissance et par la même ce qui, dans leur approche, permet de répondre à ces difficultés.

3.2.1 Difficultés relevées pour la représentation d'actions 3D pré-segmentées

Au cours de notre étude, nous avons relevé trois questions à considérer lors de la modélisation d'une action squelettique.

3.2.1.1 Comment faire face à la variabilité morphologique ?

Nous appelons *variabilité morphologique* la différence entre les données capturées pour une même classe d'action lorsqu'elle est effectuée par différents sujets. Ceci est expliqué notamment par la dépendance entre les données de capture de cette action et les caractéristiques anthropométriques du sujet l'ayant effectué. Par exemple, dans la Figure 3.2 les bras de longueurs différentes entraînent des trajectoires d'amplitudes variables, alors même que la sémantique de l'action effectuée est supposée être la même.

Or dans le domaine de la reconnaissance d'actions, la variabilité morphologique n'est pas spécifiquement prise en compte. En effet, il est souvent supposé que les descripteurs extraits sur ces données sont en mesure de neutraliser ce type de variabilité. Les seules variabilités explicitement considérées dans la littérature sont celles dues aux différences des angles de capture lorsque plusieurs caméras sont utilisées.

A l'opposé, ce problème de variabilité morphologique est central dans d'autres domaines, là où il est plus facile de ressentir les effets comme dans le domaine de l'animation

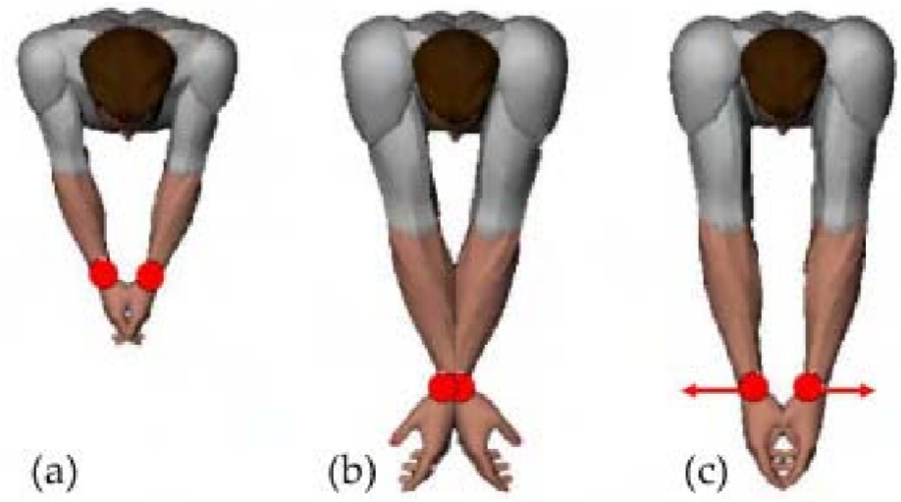

FIGURE 3.2 – Illustration du problème de morphologies différentes lors de la reconstruction, pour des besoins d'animation, de personnages virtuels à partir des données réelles [KJ05]. Les illustrations (a) et (b) mettent en avant ces problèmes lorsque des angles sont appliqués sur deux personnages dont la taille des bras ne correspond pas. L'illustration (c) montre la posture qui respecte le contact initial des mains lors de cette reconstruction.

par ordinateur. En effet, pour animer un modèle humanoïde avec les actions d'un sujet, il faudrait que cet humanoïde possède des caractéristiques morphologiques et anthropométriques identiques à celles du sujet. Comme mis en évidence sur la Figure 3.2, si on capture un mouvement d'applaudissement sur un sujet adulte pour le reconstruire sur un humanoïde de synthèse enfant en utilisant les données de capture brutes, on obtiendra nécessairement des problèmes de collision des mains.

De ce fait, nous considérons que la variabilité morphologique est une des premières difficultés à considérer lors de la conception d'une représentation d'actions 3D. Nous présenterons une solution pour remédier à cette difficulté dans l'étape de prétraitement de l'approche proposée.

3.2.1.2 Comment représenter les corrélations spatiales entre les différentes trajectoires des articulations ?

Une deuxième difficulté pour modéliser une action 3D consiste à capturer simultanément l'information de plusieurs articulations. En effet, lors de la performance d'une action 3D plusieurs articulations sont mises en jeu et la description et la modélisation des trajectoires associées servent à identifier l'action en cours (Figure 3.3). La difficulté est de concevoir des descripteurs qui permettent d'extraire suffisamment d'informations sur chacune de ces trajectoires mais aussi sur leur inter-dépendance spatiale. Nous référençons

cette deuxième difficulté comme *corrélation spatiale*.

Dans la littérature, les approches proposées modélisaient souvent ces trajectoires séparément, si bien que le nombre de descripteurs constituant les représentations proposées vient à augmenter au fur est à mesure que le nombre d'articulations capturées augmente. En effet, le nombre d'articulations suivies a doublé entre les premiers systèmes de capture et ceux disponibles aujourd'hui (allant de 15 jusqu'à plus de 30 articulations). Il devient alors nécessaire d'avoir une représentation d'actions qui tienne compte de cet ajout d'informations sans pour autant augmenter drastiquement la dimension globale de cette représentation. Un contre-exemple est donné dans la section 2.3.2.2 par Gowayyed et al. [GTHES13] sous la forme de la représentation **HOD** comportant 192 descripteurs lorsque la trajectoire d'une seule articulation est considérée alors que cette dimension devient égale à 3840 éléments lorsque tout le squelette est considéré.

En outre, certaines actions ne peuvent être caractérisées que par la connaissance simultanée des informations issues de plusieurs articulations. Une bonne représentation doit ainsi être en mesure de corréler les informations issues de ces multiples trajectoires. Nous présentons dans l'étape de conception une solution qui permet à la fois de corréler l'information spatiale des différentes trajectoires et d'avoir une représentation de taille réduite.

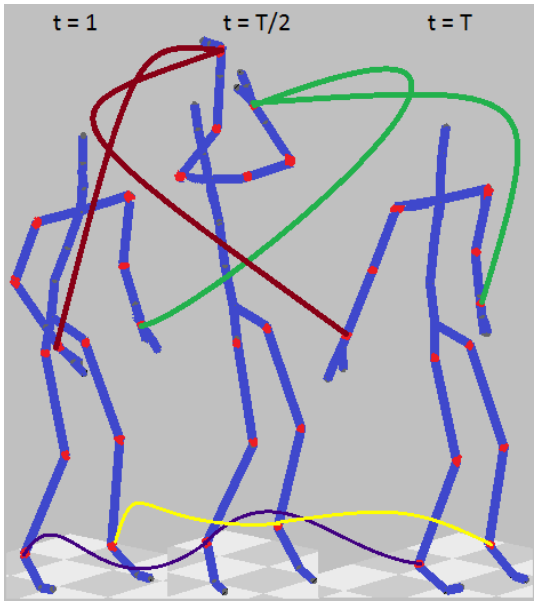

FIGURE 3.3 – Illustration de certaines trajectoires articulaires lors de la performance d'une action.

3.2.1.3 Comment représenter les dépendances temporelles intrinsèques à une action sous-tendue par plusieurs trajectoires ?

La troisième et dernière difficulté considérée lors de notre étude est appelée *séquencement temporel*. En effet, sur le plan temporel une classe d'action est définie par le séquencement des postures que prend un sujet. Au niveau des positions articulaires, ce séquencement est divisible en deux. D'une part, la succession des positions de chaque articulation et, d'autre part, la corrélation temporelle entre ces positions pour des articulations différentes. La prise en compte de cette dépendance temporelle est importante d'autant plus que certaines classes d'actions ne sont distinguables que par l'ordre dans lequel sont enchaînées les postures. Par exemple, dans un kata de karaté le fait de soulever les deux bras en même temps ou séquentiellement est associée à deux classes d'actions distinctes. Or si on modélise ces actions uniquement sur le plan spatial (c'est-à-dire en décrivant la forme globale produite lors de l'exécution de ces actions), les deux classes ne peuvent être distinguées.

Les méthodes de la littérature, qui tiennent compte de cette difficulté, utilisent dans leurs approches des classifieurs séquentiels comme les chaînes de markov (HMM) ou plus récemment les LSTM (voir section 2.3.1). Dans notre étude, nous proposons une solution alternative qui permet d'intégrer explicitement cette information dans les descripteurs extraits.

3.2.2 Approche 3DMM : *3D Multistroke Mapping*

Nous présentons dans cette section l'approche dénommée **3DMM** pour *3D Multistroke Mapping* comme une première manière d'opérer simplement le transfert des techniques de modélisation 2D pour représenter une action 3D. L'idée de base de l'approche **3DMM** est de ramener le problème de modélisation d'actions 3D dans un espace de représentation 2D. Cette approche, illustrée dans la Figure 3.4, est décrite ci-après en trois phases afin de répondre explicitement aux trois questions mises en évidence précédemment.

3.2.2.1 Réponse à la première question : prétraitement amorphologique

Cette étape de prétraitement vise principalement à répondre à la première difficulté à savoir la variabilité morphologique. Pour ce faire, nous avons retenu une solution initialement proposée dans le domaine de l'animation [HRE+08, KJ05] et qui a été évaluée par la suite en reconnaissance des gestes du haut du corps [Sor12, SKBM13]. Lors de l'évaluation en reconnaissance, il a été notamment prouvé que ce prétraitement permettait de réaliser de meilleures performances que lorsque les données brutes ou encore les données angulaires (les angles d'Euler) sont utilisées.

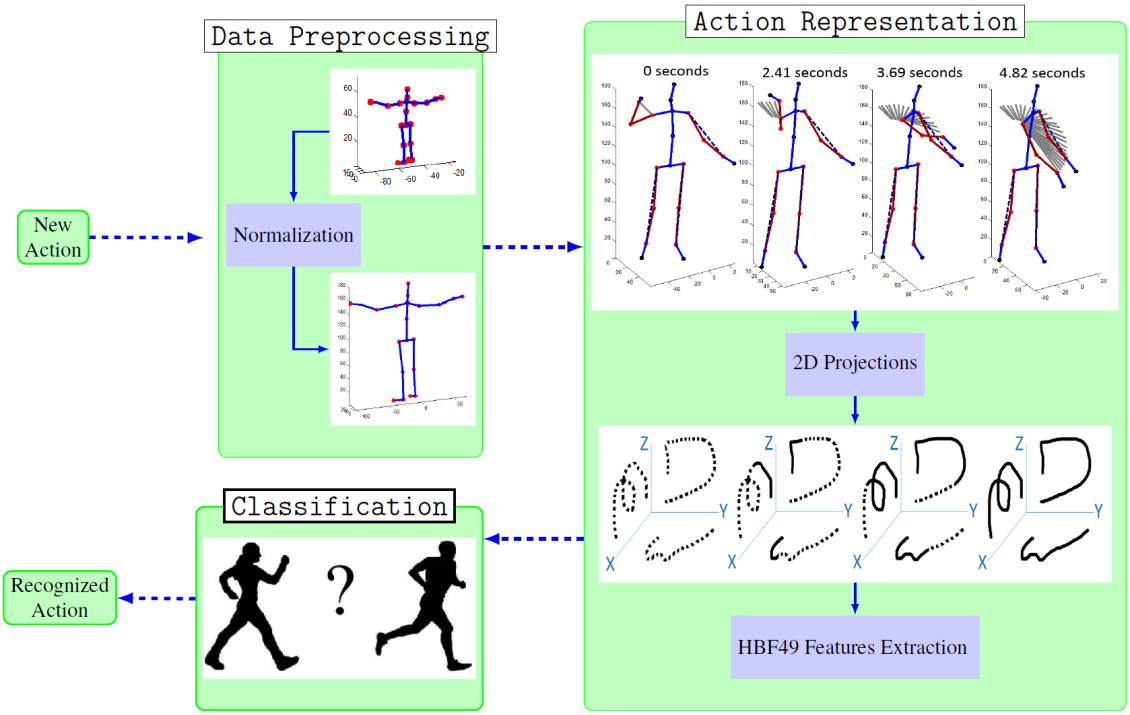

FIGURE 3.4 – Principales étapes constituant l'approche **3DMM**, proposée pour la reconnaissance des actions à base de données squelettiques.

L'idée globale de ce prétraitement étant d'agir sur les séries de données squelettiques brutes de manière à fournir de nouvelles séries de données qui seraient indépendantes de la morphologie du sujet. Globalement, ce prétraitement s'appuie sur le fait que l'information la plus importante dans un mouvement est généralement liée à la position de l'articulation distale qui est censée interagir avec les objets de l'environnement (par exemple les articulations des poignets, notées j_{PoG} et j_{PoD} dans la Figure 3.5, pour un mouvement de bras). Les articulations intermédiaires, comme les articualtions des coudes notées j_{CoG} et j_{CoD} dans la Figure 3.5, dépendent plutôt de la morphologie et du style propres à chaque sujet. Pour cette raison, la représentation amorphologique que nous proposons d'utiliser permet de neutraliser la variabilité introduite par les articulations intermédiaires.

Par exemple, pour une action corps-complet, les trajectoires 3D de douze (12) articulations associées au mouvement des bras et des jambes sont prises en compte (Figure 3.5). En particulier, les coordonnées cartésiennes des articulations suivantes sont considérées : épaules, coude et poignets pour le haut du corps et les hanches, genoux et chevilles pour le bas du corps.

La position 3D j_i^t de chaque articulation j_i à l'instant t est donnée dans un système de coordonnées centré au niveau de la hanche du squelette. Partant de ces coordonnées, quatre vecteurs sont formés suivant chaque partie du corps, à savoir, bras gauche, bras

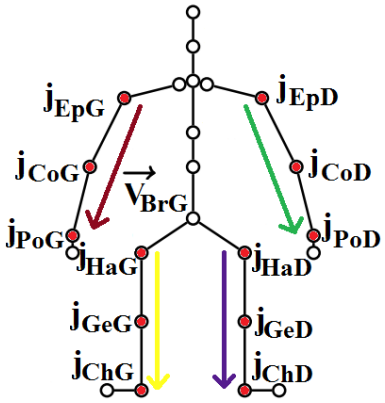

FIGURE 3.5 – illustration des articulations sélectionnées (épaules, coude et poignets pour le haut du corps et les hanches, genoux et chevilles pour le bas du corps) et des vecteurs amorphologiques associés.

droit, jambe gauche et jambe droite. Par exemple, le vecteur du bras gauche, noté $\overrightarrow{V}_{BrG}(t)$, relie les deux extrémités articulaires du bras gauche, c'est-à-dire l'épaule gauche j^t_{EpG} et le poignet gauche j^t_{PoG}, à l'instant t (Eq.3.1).

$$\vec{V}_{BrG}(t) = \overrightarrow{j^t_{EpG} j^t_{PoG}} \tag{3.1}$$

Les trois autres vecteurs, $\vec{V}_{BrD}(t)$,$\vec{V}_{JaG}(t)$ et $\vec{V}_{JaD}(t)$, sont obtenus de manière similaire. Tous ces vecteurs sont représentés sur la Figure 3.5.

Ensuite, pour encoder explicitement la variabilité morphologique, il est nécessaire de s'abstraire de la longueur de la chaîne cinématique contenant l'effecteur. Il s'en suit que ces vecteurs 3D, doivent être normalisés par leur extension maximale, c'est-à-dire par la longueur du bras ou de la jambe, respectivement. Cette normalisation est supposée réduire l'influence de la morphologie du sujet. Ainsi, le vecteur $\vec{V}_{BrG}(t)$ liant directement l'épaule au poignet est normalisé par la longueur totale du bras suivant l'équation 3.2, où $\vec{V}^{MI}_{BrG}(t)$ fait référence au vecteur amorphologique associé et j^t_{CoG} fait référence à la position du coude gauche à l'instant t.

$$\vec{V}^{MI}_{BrG}(t) = \frac{\vec{V}_{BrG}(t)}{||j^t_{EpG} j^t_{CoG}|| + ||j^t_{CoG} j^t_{PoG}||} \tag{3.2}$$

Au fur et à mesure de la performance d'une action donnée, les vecteurs ainsi définis donnent lieu à des trajectoires indépendantes de la morphologie (quatre trajectoires dans le cas d'une action corps-complet). Au lieu des données brutes fournies par le système de capture, ces trajectoires sont utilisées comme données d'entrée à l'étape de modélisation décrite ci-après.

3.2.2.2 Réponse à la deuxième question : hypothèse multistrokes

La deuxième étape de notre approche traite de la modélisation d'une action à partir des trajectoires amorphologiques produites dans l'étape précédente. Pour tirer parti du savoir-faire de l'extraction de caractéristiques de trajectoires 2D aussi bien en termes de forme de trajectoire que de corrélation spatiale, nous proposons d'extraire de ces trajectoires 3D des descripteurs initialement définis pour des trajectoires manuscrites 2D.

Nous avons opté pour l'utilisation de l'ensemble des descripteurs 2D **HBF49** [DA13], un jeu de descripteurs 2D très performant, dont les résultats expérimentaux ont montré son efficacité pour modéliser des trajectoires 2D (forme et corrélation spatiale) tout en étant de dimension très réduite (49 descripteurs seulement). Afin de rendre ce support auto-suffisant, il est nécessaire de détailler certains aspects de ce jeu de descripteurs 2D.

Tout d'abord, il est important de souligner qu'un tracé 2D ne se réfère pas seulement au texte manuscrit. Il couvre en fait un champ plus large qui, en plus de l'écriture manuscrite, comprend des diagrammes d'esquisse, des signatures, des dessins libres et des commandes de contrôle à l'aide d'un stylet. De plus en traitant de la reconnaissance de tracés manuscrits, de nombreuses taxonomies pourraient être trouvées dans la littérature en fonction de la manière dont le sujet est abordé. Nous retenons notamment celle permettant de distinguer entre les représentations sur la base du nombre de traits, dits aussi **stroke**, qui composent un tracé. Une stroke est une trace du mouvement de la pointe d'un stylet qui débute lorsque cette pointe est au contact d'une surface et s'achève lorsqu'elle est relevée.

Les premières représentations de tracés 2D se focalisaient sur la modélisation et la reconnaissance de tracés à une seule stroke dits **tracés monostrokes**. Le système de reconnaissance dénommé *one-Dollar* proposé par Wobbrock et al. [WWL07] est un exemple très connu d'approche de reconnaissance de tracés monostrokes 2D. A l'opposé, un système de reconnaissance de tracés **multistrokes** doit prendre en compte plusieurs paramètres dont les variations de forme, les différences dans l'ordre des strokes, un nombre variable de strokes. La Figure 3.6 illustre cette variation pour une lettre "E".

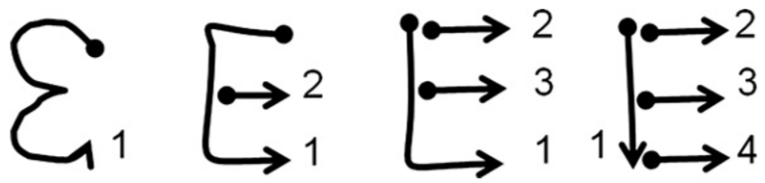

FIGURE 3.6 – Illustration de la variation du nombre de strokes pour la lettre "E" (de un à quatre strokes). La direction de la stroke est indiquée par un point au début d'un trait, et l'ordre du trait par le nombre à la fin d'un trait [MTI10].

Notre choix de se baser sur le jeu de descripteurs HBF49 est motivé par au moins trois raisons. La première est que ce jeu de descripteur a été conçu de manière à être le plus générique possible, c'est-à-dire peut représenter des patterns issus de contextes variés. La deuxième raison est qu'il regroupe, en un jeu très réduit, des descripteurs caractérisant aussi bien l'information globale relative à la forme du pattern mais aussi l'information de corrélation entre les différents tracés. Par exemple, le volume de la boite englobant le tracé 2D (Figure 3.7-a) est un des descripteurs permettant de le caractériser d'un point de vue global, alors que le regroupement (comptage) des points 2D de chaque stroke sous forme d'un histogramme (Figure 3.7-b) permet de caractériser les corrélations entre les différentes strokes composant ce tracé. La troisième raison est que ce jeu de descripteurs est en mesure de modéliser des tracés aussi bien monostroke que multistroke avec le même nombre réduit de descripteurs. Autrement dit, il n'y a aucune dépendance entre le nombre de descripteurs et le nombre de strokes composant le tracé à modéliser.

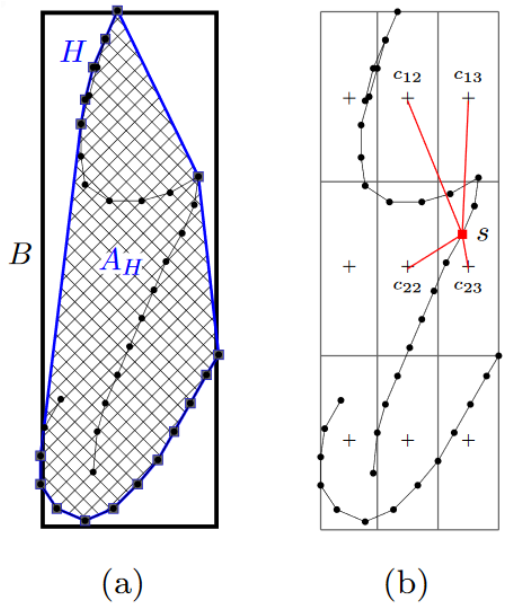

FIGURE 3.7 – Illustration d'un symbole formé par deux strokes pour lesquels **(a)** le volume de la boite englobante et **(b)** l'histogramme comptabilisant la répartition des points sont calculés [DA13].

S'agissant de la modélisation d'une action 3D squelettique, une approche "naïve" est de projeter les trajectoires 3D dans trois plans orthogonaux et ensuite établir un parallèle entre les trajectoires 2D ainsi obtenues et des tracés manuscrits classiques. Comme illustré sur la Figure 3.8, si l'on considère pour simplifier les trajectoires des 2 jambes et des 2 bras, cette projection produit quatre tracés sur chacun des trois plans. Des tracés 2D sont ainsi obtenus sur chacun des plans et il est alors possible d'en extraire un ensemble de

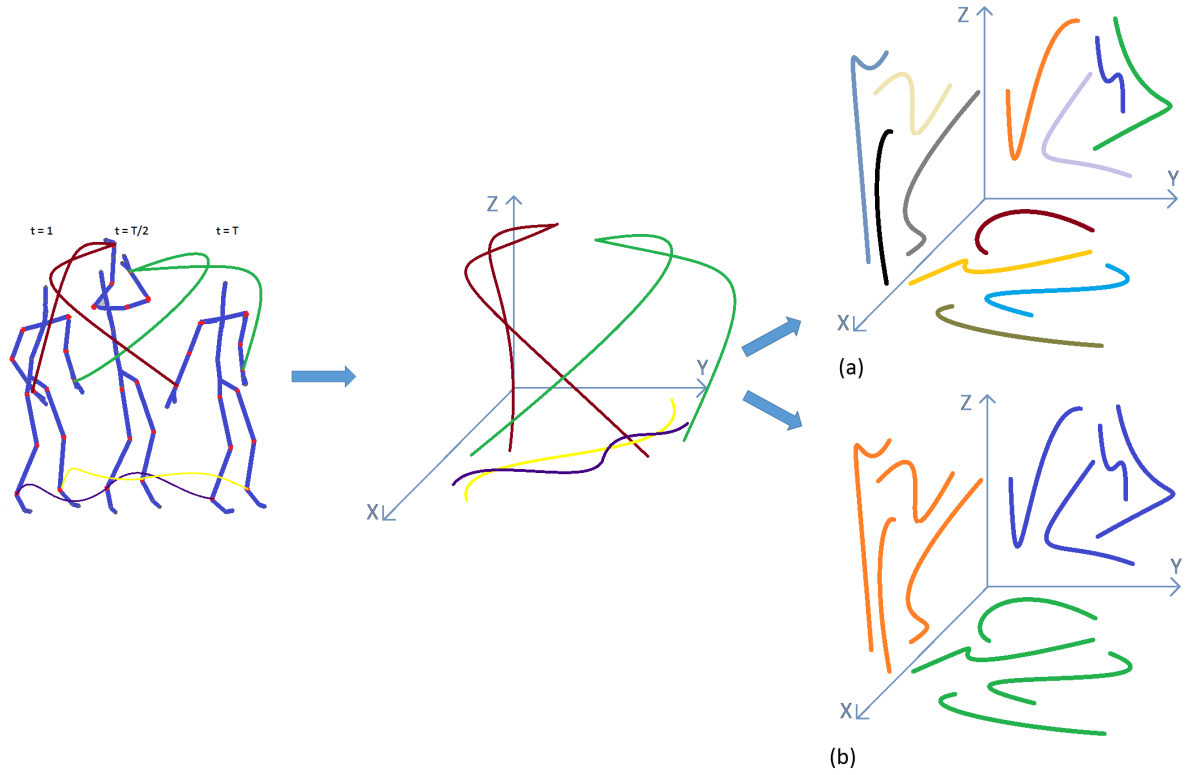

FIGURE 3.8 – Pour permettre l'utilisation de descripteurs 2D, chacune des trajectoires amorphologiques est projetée sur les trois plans. **(a)** Dans la version **monostroke**, chaque trajectoire est considérée indépendamment des autres et est assimilée à un tracé 2D composé d'un seul trait, d'où la couleur différente pour chacune de ces trajectoires. **(b)** Dans la version **multistrokes**, un tracé est composé de toutes les trajectoires projetées appartenant au même plan, et est coloré de la même manière dans cette illustration.

descripteurs 2D pour les caractériser.

Pour extraire les descripteurs HBF49 à partir des trajectoires d'actions projetées, nous avons deux stratégies. En particulier, dans chacun des trois plans, l'extraction de descripteurs peut être effectuée soit en considérant que les strokes sont séparées et que le tracé projeté est en réalité quatre **pseudo-symbole monotroke** (Figure 3.8a) ou bien en considérant le tracé projeté comme un **pseudo-symbole multistrokes** (Figure 3.8b).

Nous avons opté pour une modélisation suivant la stratégie multistrokes pour deux raisons principales. D'une part, cette stratégie permet de tenir compte de la corrélation spatiale entre les différentes strokes (les trajectoires de différents membres du corps). D'autre part, elle permet d'avoir une représentation de taille réduite. En effet, en extrayant 49 descripteurs sur chacun des pseudo-symboles projetés et en concaténant tous ces descripteurs pour former une représentation de l'action, on obtient une dimension égale à

147 (49 * 3 plans). Or avec une stratégie monostroke, il est nécessaire d'extraire 196 (49 * 4 strokes) descripteurs suivant chaque plan, ce qui résulte au final en 588 (196 * 3 plans) descripteurs.

Nous considérons que l'extension du concept multistrokes pour gérer simultanément plusieurs trajectoires de squelette est une approche simple mais originale qui donnera de bons résultats (cf. section 3.4). Elle permet, entre autre, d'extraire l'information de corrélation spatiale et donc d'adresser la deuxième question mise en évidence lors de notre étude. Ce choix a principalement justifié l'appellation de *3DMM : 3D Multistroke Mapping* donnée à notre approche de reconnaissance.

3.2.2.3 Réponse à la troisième question : hiérarchie temporelle

La représentation telle que présentée jusque-là ne tient pas compte de la dépendance temporelle au sein d'une séquence. En réalité, les descripteurs HBF49 [DA13] ne capturent que l'information spatiale relative à la forme globale produite par une action. Or comme mis en évidence dans la section 3.2.1.3, il est important de caractériser, d'une manière explicite, les dépendances temporelles entre les trajectoires sous-tendues par une action. Nous avons alors construit notre représentation suivant une division temporelle à plusieurs niveaux de la séquence.

Cette manière d'intégrer la temporalité est couramment adoptée dans de nombreuses représentations d'actions en utilisant à chaque fois différentes variantes comme dans les travaux de [HTGES13, ESH14, GTHES13] (cf. section 2.3.2). Pour nos travaux, nous avons adopté une variante légèrement différente (Figure 3.9). Il s'agit d'abord d'extraire les descripteurs sur la totalité de la séquence suivant le schéma multistrokes défini ci-dessus. L'objectif est de produire une vision complète des formes et des dépendances spatiales entre les trajectoires. Cette première opération d'extraction de descripteurs est illustrée au premier niveau de la Figure 3.9 et permet d'obtenir les 147 premiers descripteurs. Ensuite, en suivant le même schéma multistrokes, les descripteurs sont extraits pour trois sous-séquences relatives au début, au milieu et à la fin de l'action. Comme illustré sur le deuxième niveau de la Figure 3.9, nous imposons à ce que ces sous-séquences se chevauchent sur un tiers de leur longueur de manière à assurer une modélisation continue de la séquence. L'objectif de ce deuxième niveau d'extraction est de capturer la dépendance spatiale locale relativement à une fenêtre temporelle.

Nous illustrons sur la Figure 3.10 le pseudo-symbole multistrokes considéré sur le premier niveau et ceux considérés sur les trois fenêtres temporelles du deuxième niveau. Cette illustration met en avant le fait que la division temporelle s'opère sur la séquence d'action et pas sur le pseudo-symbole multistroke formé. A l'issue de cette étape, nous obtenons 441 (147×3) nouveaux descripteurs. Pour notre approche nous avons limité

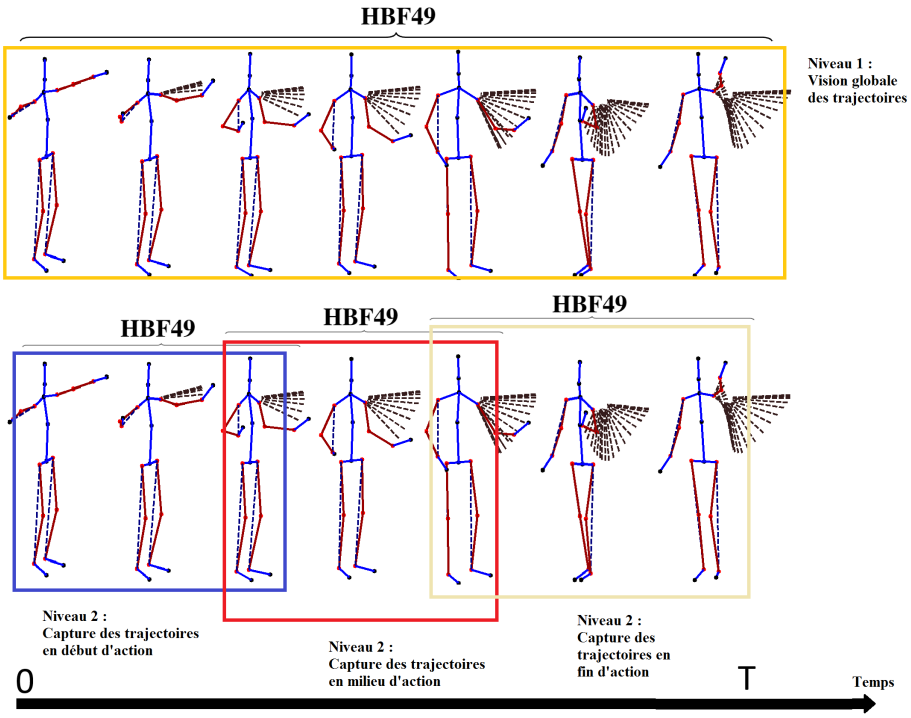

FIGURE 3.9 – Illustration du partitionnement temporel adopté dans notre représentation. L'extraction des descripteurs au $i^{ème}$ niveau couvre les $\frac{T}{3^i}$ frames de la séquence, où T est la longueur de la séquence entière.

le nombre de niveaux à deux, ce qui porte la dimension finale de la représentation à 588 ($147 \times 1 + 147 \times 3$).

Enfin, vu que nous extrayons les mêmes descripteurs suivants les 3 plans de projection, une certaine redondance est présente et peut détériorer les performances de reconnaissance. Pour optimiser cette représentation, une sélection parmi les 588 descripteurs est opérée au moyen d'un algorithme de sélection assez répandu, à savoir One-R [Hol93].

Dans cette étude, nous nous focalisons davantage sur la conception d'une nouvelle représentation d'actions 3D et la mesure de l'impact des descripteurs plutôt que sur le moteur de classification. Ainsi, nous avons choisi d'utiliser des machines à vecteurs de support (SVM) lors de l'étape de classification. La configuration de base adoptée est un noyau polynomial avec des paramètres optimisés expérimentalement au moyen d'une grille de recherche sur un ensemble de validation. L'implémentation utilisée de ce classifieur est celle fournie dans LIBSVM [CL11a]. Nous présentons les résultats obtenus dans la section 3.4.

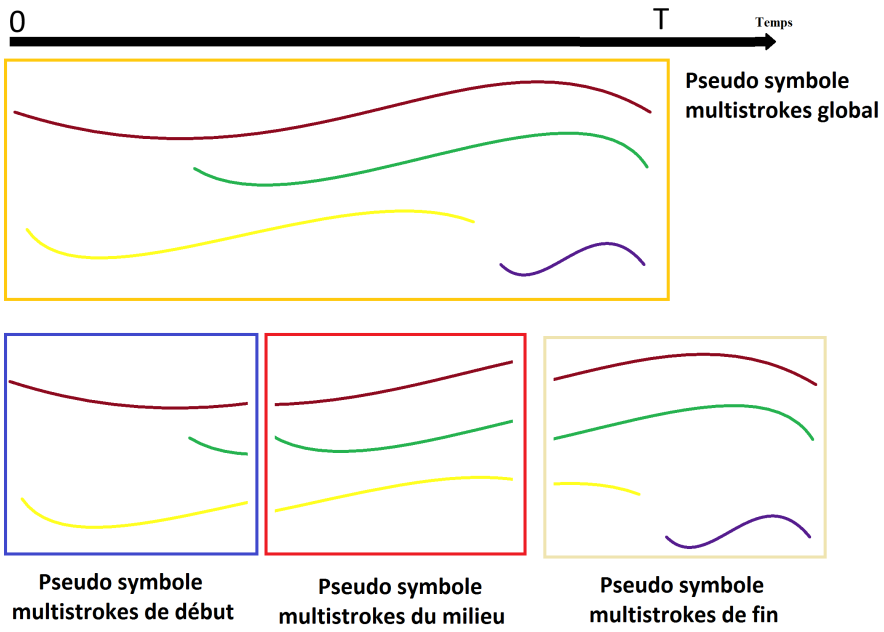

FIGURE 3.10 – Illustration des trajectoires formant les pseudo-symboles multistrokes global (niveau 1), de début, du milieu et de fin (niveau 2).

3.3 Transfert d'un jeu de descripteurs 2D à l'espace de représentation d'actions 3D : jeu de descripteurs HIF3D

La seconde approche que nous avons mise en place consiste à concevoir un nouveau jeu de descripteurs calculés directement dans l'espace 3D (sans projection) pour représenter des actions 3D en s'inspirant des descripteurs 2D. Cette deuxième proposition fait suite à une analyse de l'approche **3DMM** présentée dans la section 3.2.

En effet, le fait de modéliser un pattern produit dans un environnement 3D (c'est-à-dire une action 3D) en utilisant des projections dans des dimensions inférieures (c'est-à-dire des plans 2D) engendre nécessairement une perte d'informations. De plus, extraire les mêmes descripteurs suivant les différents plans introduit une redondance qui peut nuire aux performances globales. Ceci est vrai en dépit du fait que cette redondance est en partie supprimée par l'étape de sélection des descripteurs. Enfin, la sélection des descripteurs dépend des classes considérées. Elle doit donc être opérée pour chaque nouvelle base de données, voire même pour chaque protocole de test. Cette sélection est donc problématique car elle ne permet pas d'avoir une seule représentation globale (cf. section 3.4).

Afin de pallier ces inconvénients, nous proposons d'opérer le transfert 2D-3D en conce-

vant un nouveau jeu de descripteurs que nous dénommons **HIF3D** pour *Handwriting-Inspired Features for 3D skeleton-based action recognition*. **HIF3D** est en fait une extension aux actions 3D squelettiques du jeu de descripteurs 2D HBF49 [DA13], que nous avons présenté en section 3.2.2.2 et utilisé dans l'approche **3DMM**. A la différence de ce qui a été fait pour l'approche **3DMM**, les trajectoires amorphologiques formées (cf. section 3.2.2.1) sont rapportées dans un repère 3D ayant pour origine le centre articulaire de la hanche. Comme illustré dans la Figure 3.11, il en résulte un pattern 3D composé de quatre strokes, c'est-à-dire un pseudo-symbole multistrokes 3D, à partir duquel les descripteurs **HIF3D** sont extraits.

Dans ce qui suit, nous présentons en détail le jeu de descripteurs **HIF3D**. Nous introduisons dans une première section une série de notations. Ensuite, nous présentons le premier sous-ensemble de descripteurs, nommés **descripteurs étendus**. Il s'agit des descripteurs issus d'une adaptation 3D directe des descripteurs contenus dans HBF49 [DA13]. Enfin, nous décrivons le second sous-ensemble de descripteurs, nommés les **descripteurs inspirés**. Ce sont des descripteurs nouvellement introduits, formulés différemment et qui permettent d'extraire pour des patterns 3D les mêmes informations que celles extraites par leur équivalents 2D.

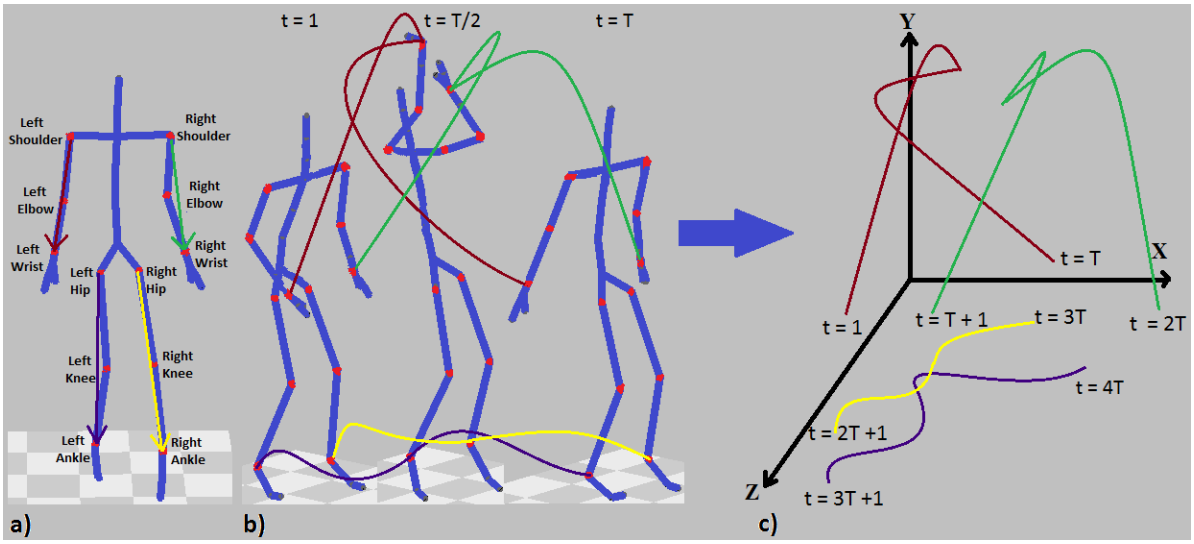

FIGURE 3.11 – (a) Illustration avec K=4 vecteurs formés à partir des douze articulations sélectionnées. (b) Illustration de la progression des quatre vecteurs amorphologiques. (c) Assemblage *temporel* des quatre trajectoires amorpholgiques pour former un pattern multistrokes 3D.

3.3.1 Notations

- Un pattern S est une séquence de points 3D résultant de l'assemblage de K trajectoires amorphologiques (illustration avec K=4 trajectoires pour une action corps-complet dans la Figure 3.11). $S = \{s_1, ..., s_T, s_{T+1}, ..., s_{2T}, s_{2T+1}, ..., s_{(K-1)T}, s_{(K-1)T+1}, ..., s_n\}$, où T est la longueur de chaque trajectoire (ou stroke) et $n = K \times T$ est le nombre de points dans S. Chaque point $s_i = (x_i, y_i, z_i)$ est situé dans l'espace tridimensionnel,
- s_1 et s_n sont respectivement le premier et le dernier point de S,
- $||.||$ dénote la distance euclidienne entre les points,
- $L = L_{1,n}$ est la longueur totale de S,
- s_m est le point situé au milieu du pattern,
- x_{max} est l'abscisse du point le plus à droite de S, x_{min}, y_{max}, y_{min}, z_{max} et z_{min} sont les coordonnées des autres extrémités,
- B est la boîte englobante de S : c'est-à-dire le parallélépipède défini par les axes x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max},
- $\mathbf{w} = x_{max} - x_{min}$ est la largeur de B, $\mathbf{h} = y_{max} - y_{min}$ est la hauteur de B, et $\mathbf{d} = z_{max} - z_{min}$ est la profondeur de B, (si $\mathbf{w}$, $\mathbf{h}$ ou $\mathbf{d}$ sont nulles, leur valeur est mise à 1),
- $l = max(\mathbf{w}, \mathbf{h}, \mathbf{d})$,
- c_x, c_y et c_z sont les coordonnées du centre de B,
- $\mu(\mu_x, \mu_y, \mu_z) = (1/n) \sum_{i=1}^{n} s_i$ est le centre de gravité du pattern S.

3.3.2 Premier sous-ensemble : les descripteurs étendus

Tout d'abord, il est important de préciser que le jeu HBF49 sur lequel nous nous basons n'est pas issu d'une simple concaténation de descripteurs choisis arbitrairement parmi ceux retrouvés dans la littérature 2D. Au contraire, cet ensemble a été conçu suivant une approche constructive de manière à caractériser de façon générique la forme d'un pattern ainsi que les dépendances spatiales entre les strokes qui le composent. En particulier, les auteurs ont d'abord identifié les différents aspects relatifs à la forme globale d'un pattern et aux dépendances entre ses strokes et ont retenu, pour chaque aspect, le descripteur ou le groupe de descripteurs à même de le caractériser. C'est en ce sens que le savoir-faire 2D et l'effort qui a été déjà fait sont importants et peuvent être étendus pour caractériser des patterns composés de strokes 3D.

En ce qui concerne notre jeu de descripteurs, le premier sous-ensemble regroupe les descripteurs ayant été directement étendus aux patterns 3D. Nous recensons un total de 41 descripteurs décrits comme suit.

Les points de départ et de fin : Les positions du premier et du dernier point, illustrés

sur la Figure 3.12, constituent des descripteurs importants pour distinguer les patterns dans de nombreuses situations. Par exemple, les gestes simples présentent souvent des points de début et de fin stables.

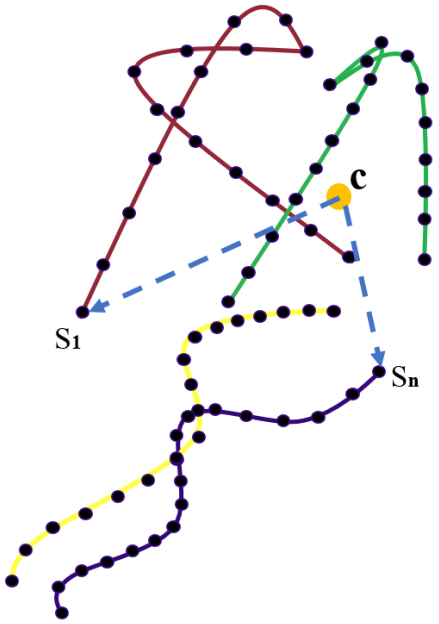

FIGURE 3.12 – Illustration des vecteurs de départ et de fin pour un pattern à quatre strokes.

Nous calculons les coordonnées par rapport à un cuboïde virtuelle de côté $l = max(\mathbf{w}, \mathbf{h}, \mathbf{d})$, centré au centre $c = c_x, c_y, c_z$, de manière à éviter les mesures bruitées dans le cas de patterns de très faible dimension. Les deux descripteurs du point de départ sont calculés comme suit :

$$\mathbf{f_1} = \frac{x_1 - c_x}{l} + \frac{1}{2}, \qquad \mathbf{f_2} = \frac{y_1 - c_y}{l} + \frac{1}{2}, \qquad \mathbf{f_3} = \frac{z_1 - c_z}{l} + \frac{1}{2} \tag{3.3}$$

De façon similaire les points de fin s'obtiennent :

$$\mathbf{f_4} = \frac{x_n - c_x}{l} + \frac{1}{2}, \qquad \mathbf{f_5} = \frac{y_n - c_y}{l} + \frac{1}{2}, \qquad \mathbf{f_6} = \frac{z_n - c_z}{l} + \frac{1}{2} \tag{3.4}$$

Vecteur reliant les premier et dernier points : Le vecteur $\vec{v} = \overrightarrow{s_1 s_n}$, illustré sur la Figure 3.13, comporte des informations supplémentaires sur la dynamique du pattern. Ainsi, nous mesurons la longueur du vecteur $||\vec{v}|| = ||s_1 s_n||$, ainsi que le cosinus et le sinus de son angle par rapport à la ligne horizontale.

$$\mathbf{f_7} = ||\vec{v}||, \quad \mathbf{f_8} = \frac{\vec{v}.\vec{u_x}}{||\vec{v}||}, \quad \mathbf{f_9} = \frac{\vec{v}.\vec{u_y}}{||\vec{v}||}, \quad \mathbf{f_{10}} = \frac{\vec{v}.\vec{u_z}}{||\vec{v}||} \tag{3.5}$$

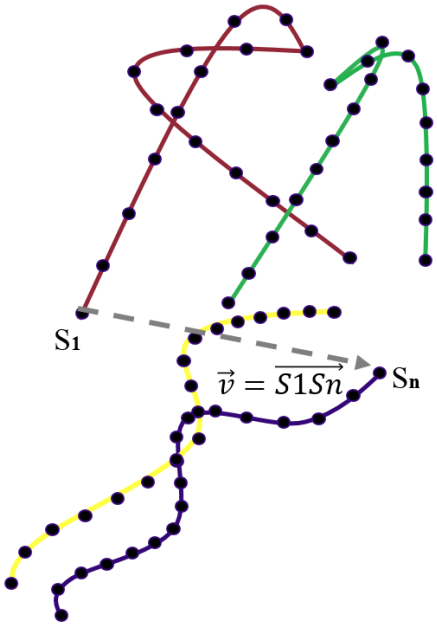

FIGURE 3.13 – Illustration du vecteur reliant les premier et dernier points.

Fermeture : Elle permet de mettre en évidence les différences entre l'aspect fermé ou allongé d'un pattern. Il est défini comme :

$$\mathbf{f_{11}} = \frac{||\vec{v}||}{L} \tag{3.6}$$

Angle du vecteur initial : Le vecteur initial est défini par les premiers points de la trajectoire. Dans notre implémentation nous considérons en fait le vecteur entre les premier et troisième points : $\vec{w} = \overrightarrow{s_1 s_3}$ (cf. Figure 3.14). L'angle initial est décrit par les mesures cosinus et sinus :

$$\mathbf{f_{12}} = \frac{\vec{w}.\vec{u_x}}{||\vec{w}||}, \quad \mathbf{f_{13}} = \frac{\vec{w}.\vec{u_y}}{||\vec{w}||}, \quad \mathbf{f_{14}} = \frac{\vec{w}.\vec{u_z}}{||\vec{w}||} \tag{3.7}$$

Inflexions : Trois descripteurs d'inflexion sont calculés au moyen du point médian s_m du pattern S et du point médian du segment $s_1 s_n$. Ces descripteurs servaient à la base à distinguer entre les patterns sous forme d'arcs ayant des orientations différentes.

$$\mathbf{f_{15}} = \frac{1}{\mathbf{w}}(x_m - \frac{x_1 + x_n}{2}), \quad \mathbf{f_{16}} = \frac{1}{\mathbf{h}}(y_m - \frac{y_1 + y_n}{2}), \quad \mathbf{f_{17}} = \frac{1}{\mathbf{d}}(z_m - \frac{z_1 + y_n}{2}) \tag{3.8}$$

Proportion des segments descendants : Comme le démontrent Anquetil et Lorette [AL97], les parties de trajectoires orientées vers le bas de la surface d'écriture (c'est-à-dire orientées vers des valeurs croissantes en dimension y) sont particulièrement importantes

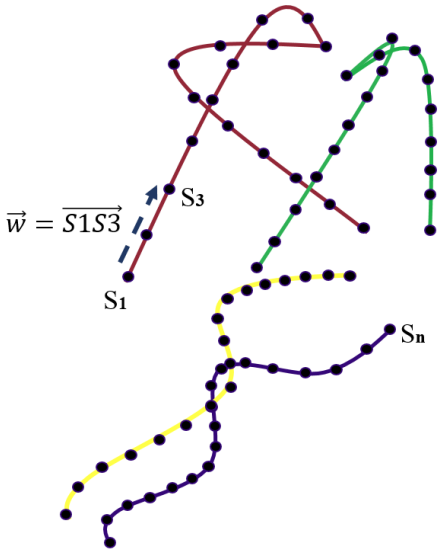

FIGURE 3.14 – Illustration du vecteur reliant les premier et dernier points.

dans la perception de l'écriture. En l'adaptant pour le cas 3D, cette information est exprimée par la proportion de longueur de segments orientés vers une des trois direction X, Y ou Z, et se calcule comme suit :

$$\mathbf{f_{18}} = \sum_{k=1}^{p_x} LX_k, \quad \mathbf{f_{19}} = \sum_{k=1}^{p_y} LY_k, \quad \mathbf{f_{20}} = \sum_{k=1}^{p_z} LZ_k \tag{3.9}$$

Avec p_x, p_y, p_z le nombre de segments orientés dans le sens positif et LX_k, LY_k, LZ_k leur longueur le long des axes X, Y et Z.

Angle diagonal de la boîte englobante : Nous mesurons les trois ratios de chacun des côtés de la boîte englobante illustrée dans la Figure 3.15, comme suit :

$$\mathbf{f_{21}} = \arctan\left(\frac{\mathbf{h}}{\mathbf{w}}\right), \quad \mathbf{f_{22}} = \arctan\left(\frac{\mathbf{d}}{\mathbf{h}}\right), \quad \mathbf{f_{23}} = \arctan\left(\frac{\mathbf{w}}{\mathbf{d}}\right) \tag{3.10}$$

Longueur de la trajectoire : Ces descripteurs rapportent une information indépendante de l'orientation :

$$\mathbf{f_{24}} = L, \tag{3.11}$$

Complexité graphique : Le rapport entre le demi-périmètre de la boîte englobante et la longueur de la trajectoire permet d'avoir une information indépendante de la taille du pattern et se calcule comme suit :

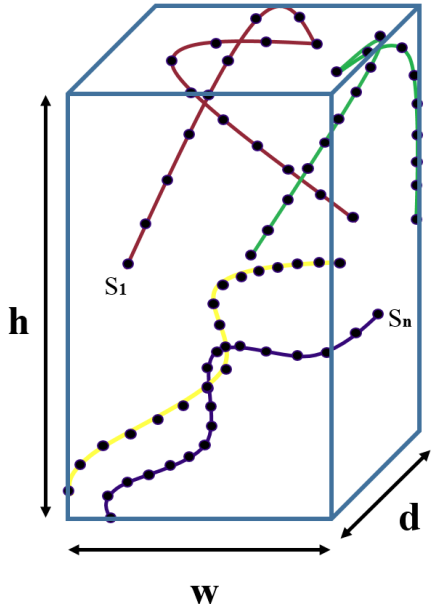

FIGURE 3.15 – Illustration de la boite englobante du pattern 3D, avec une hauteur **h**, une largeur **w** et une profondeur **d**.

$$\mathbf{f_{25}} = \frac{\mathbf{w} + \mathbf{h} + \mathbf{d}}{L} \tag{3.12}$$

Ce descripteur peut en fait caractériser la complexité graphique des actions, par exemple différencier entre une rotation simple et une rotation multiple des bras ayant la même boîte englobante.

Déviation : Ceci est un autre descripteur indépendant de l'orientation qui évalue la distance moyenne entre les points du pattern S et le centre de gravité μ :

$$\mathbf{f_{26}} = \frac{1}{n}\sum_{i=1}^{n} ||\overrightarrow{s_i\mu}|| \tag{3.13}$$

Direction moyenne : Ces descripteurs extraient une information directionnelle en calculant la moyenne des directions des segments définis dans la trajectoire de S, deux par deux :

$$\begin{gathered}\mathbf{f_{27}} = \frac{1}{n-1}\sum_{i=1}^{n-1} \arctan\left(\frac{x_{i+1}-x_i}{z_{i+1}-z_i}\right), \quad \mathbf{f_{28}} = \frac{1}{n-1}\sum_{i=1}^{n-1} \arctan\left(\frac{y_{i+1}-y_i}{x_{i+1}-x_i}\right), \\ \mathbf{f_{29}} = \frac{1}{n-1}\sum_{i=1}^{n-1} \arctan\left(\frac{z_{i+1}-z_i}{y_{i+1}-y_i}\right)\end{gathered} \tag{3.14}$$

Histogramme d'angles absolus : Ces descripteurs sont basés sur un histogramme d'angles "absolus" à huit valeurs d'entrée discrètes (h_1-h_8). Il permet de faire le recensement du nombre de segments orientés dans huit directions. Pour chaque segment, l'orientation est calculée par :

$$\alpha_i = \arccos\left(\frac{x_{i+1} - x_i}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}\right) \tag{3.15}$$

Chaque angle α_i se situe entre deux directions consécutives par exemple $\vec{v_1}$ (de valeur h_1) et $\vec{v_2}$ (de valeur h_2). La quantification se fait suivant un comptage flou, c'est-à-dire qu'un angle contribue à deux directions distinctes (contrairement à la quantification directe où seule la valeur d'entrée la plus proche est incrémentée). Supposons que α_i est plus proche de $\vec{v_1}$ que $\vec{v_2}$, alors il contribue à h_1 et h_2 avec les poids w_1 et w_2 :

$$w_1 = 1 - \frac{\angle(\vec{u_{\alpha_i}}, \vec{v_1})}{\pi/4}, \qquad w_2 = 1 - w_1 \tag{3.16}$$

Où $\vec{u_{\alpha_i}}$ est le vecteur unitaire orienté par α_i. Enfin, les quatre descripteurs $\mathbf{f_{30}}$ - $\mathbf{f_{33}}$ sont calculés comme la somme des contributions de tous les angles α_i à chacune des directions opposées, afin de garantir l'indépendance par rapport à la direction de la performance du geste.

$$\mathbf{f_{30}} = \frac{h_1 + h_5}{n}, \quad .. \quad \mathbf{f_{33}} = \frac{h_4 + h_8}{n} \tag{3.17}$$

Nous obtenons huit autres descripteurs, à savoir $\mathbf{f_{34}}$ - $\mathbf{f_{37}}$ et $\mathbf{f_{38}}$ - $\mathbf{f_{41}}$ en suivant la procédure précédente et en substituant respectivement les couples de coordonnées (x_i, y_i) avec (y_i, z_i) et (z_i, x_i) dans la formule 3.15.

3.3.3 Second sous-ensemble : les descripteurs inspirés

Le deuxième sous-ensemble de descripteurs permet d'extraire aussi l'information caractéristique identifiée comme telle dans le domaine de la reconnaissance 2D. Néanmoins, la formulation mathématique de ces descripteurs est radicalement différente de celle utilisée en 2D puisque les formules 2D originales ne peuvent pas être directement appliquées pour le cas 3D. Nous exprimons ci-après les 48 descripteurs restants.

Courbure et perpendicularité : Ces descripteurs sont indépendants de l'orientation et permettent de cumuler la courbure locale et de mesurer la perpendicularité entre les segments et les plans dans le pattern S. Tout d'abord, nous notons θ_i l'angle défini par des segments consécutifs dans la même stroke :

$$\theta_i = \arccos\left(\frac{\overrightarrow{s_{i-1}s_i}.\overrightarrow{s_is_{i+1}}}{||\overrightarrow{s_{i-1}s_i}||||\overrightarrow{s_is_{i+1}}||}\right) \tag{3.18}$$

La courbure et la perpendicularité sont définies comme :

$$\mathbf{f_{42}} = \sum_{i=2}^{n-1} \theta_i, \quad \mathbf{f_{43}} = \sum_{i=2}^{n-1} \sin^2(\theta_i) \tag{3.19}$$

La valeur de la courbure pour une ligne droite est nulle, tandis qu'elle est élevée pour des formes fortement courbées. De plus, le descripteur basé sur la perpendicularité permet de détecter des changements brusques de direction dans la trajectoire.

Nous obtenons deux autres descripteurs $\mathbf{f_{44}}$ et $\mathbf{f_{45}}$ en substituant θ_i dans la formule 3.19 avec ϕ_i qui est l'angle défini par des plans consécutifs π au sein d'une même stroke (formule 3.20).

$$\phi_i = \angle(\pi_{i-1,i,i+1}, \pi_{i,i+1,i+2}) \tag{3.20}$$

k-perpendicularité et k-angle : En introduisant un paramètre k, d'autres descripteurs angulaires indépendants de l'orientation sont définis pour constituer une autre mesure des angles locaux. L'angle précédent θ_i est étendu à θ_i^k :

$$\theta_i^k = \arccos\left(\frac{\overrightarrow{s_{i-k}s_i}.\overrightarrow{s_is_{i+k}}}{||\overrightarrow{s_{i-k}s_i}||||\overrightarrow{s_is_{i+k}}||}\right) \tag{3.21}$$

Où s_{i-k} et s_{i+k} doivent appartenir à la même stroke. A partir des angles θ_i^k, nous calculons la k-perpendicularité et le k-angle maximal :

$$\mathbf{f_{46}} = \sum_{i=k+1}^{n-k} \sin^2(\theta_i^k), \quad \mathbf{f_{47}} = \max_{i=k+1}^{n-k} \theta_i^k \tag{3.22}$$

De même, nous obtenons les descripteurs $\mathbf{f_{48}}$ et $\mathbf{f_{49}}$ en étendant l'angle ϕ_i à ϕ_i^k et en le remplaçant dans la formule 3.22 :

$$\phi_i^k = \angle(\pi_{i-k,i,i+k}, \pi_{i,i+k,i+2k}) \tag{3.23}$$

Pour nos expériences, nous avons fixé k à 2, similairement à ce qui est fait en 2D [DA13].

Histogramme d'angles relatifs : Un autre histogramme directionnel permet de mesurer les changements locaux de direction, en complément aux descripteurs basés sur la courbure ou la perpendicularité. Il s'agit aussi de descripteurs indépendants de l'orientation du pattern. Pour ce faire, les angles locaux relatifs sont d'abord calculés, en opérant une combinaison linéaire de θ_i et θ_i^k : (voir les définitions dans les équations 3.18 et 3.21)

$$\psi_i^k = \gamma\theta_i + (1-\gamma)\theta_i^k \tag{3.24}$$

Nous retenons les mêmes valeurs empiriques de $\gamma = 0.25$ et $k = 2$ comme celles utilisées dans HBF49 par [DA13]. Les contributions des angles ψ_i^k sont ensuite cumulées dans un histogramme à quatre valeurs discrètes uniformément distribuées entre $[0, \pi]$. Comme pour l'histogramme des angles absolus (cf. formule 3.16), les contributions à l'histogramme des angles relatifs sont pondérées par l'inverse de leur distance angulaire par rapport aux deux directions voisines. Enfin, quatre descripteurs $\mathbf{f_{50}}$ - $\mathbf{f_{53}}$ sont calculés à partir des valeurs de l'histogramme divisées par n.

De la même manière, nous extrayons quatre autres descripteurs $\mathbf{f_{54}}$ - $\mathbf{f_{57}}$ en considérant les angles χ_i^k obtenus de ϕ_i et ϕ_i^k comme suit :

$$\chi_i^k = \gamma\phi_i + (1-\gamma)\phi_i^k \tag{3.25}$$

Histogramme de zoning 3D flou : Nous définissons une partition régulière de la boîte englobante B en $3 \times 3 \times 3$ voxels (cubes), qui permet de fournir une description globale de la répartition des points. Ceci résulte en vingt-sept descripteurs de zoning 3D. Un histogramme est utilisé pour comptabiliser la contribution de chaque point aux huit voxels voisins qui l'entourent. Comme pour les autres histogrammes, la contribution est pondérée de façon floue, où les poids dépendent de la distance entre le point et les centres $c_{j,k,l}$, $1 \leq j, k, l \leq 3$, des voxels.

La Figure 3.16 illustre la partition de centres $c_{j,k,l}$, $1 \leq j, k, l \leq 3$, définie sur la boîte englobante d'un pattern d'une action. Pour chaque point s_i, la distance qui le sépare des centres $c_{j,k,l}$ est calculée. Ces distances sont utilisées pour calculer les poids $\mu_{ijk}(s_i)$ correspondant à la contribution du point s_i au voxel de centre $c_{j,k,l}$. La somme de ces poids pour chaque point est égale à 1 de manière à ce que plus ce point est proche du centre d'un voxel et plus le poids associé est important. Les descripteurs $\mathbf{f_{58}}$ - $\mathbf{f_{84}}$ rapportent la contribution cumulée des points de S aux vingt-sept voxels, divisée par n :

$$\mathbf{f_{58}} = \frac{1}{n}\sum_{i=1}^{n}\mu_{111}(s_i), .. \quad \mathbf{f_{84}} = \frac{1}{n}\sum_{i=1}^{n}\mu_{333}(s_i) \tag{3.26}$$

Avec $0 \leq \mu_{jkl}(s_i) \leq 1$ la contribution de chaque point s_i au voxel de centre $c_{j,k,l}$ pour $1 \leq j, k, l \leq 3$

Invariants de moments 3D : Vu que les moments de Hu sont définis spécifiquement pour des symboles 2D (des images), nous avons adopté d'autres invariants, propres à des trajectoires 3D [SH80]. Ils permettent de construire des descripteurs invariants aux déformations telles que la rotation, la translation ou le zoom. Pour ce faire, nous calculons d'abord les moments centraux d'inertie en 3D :

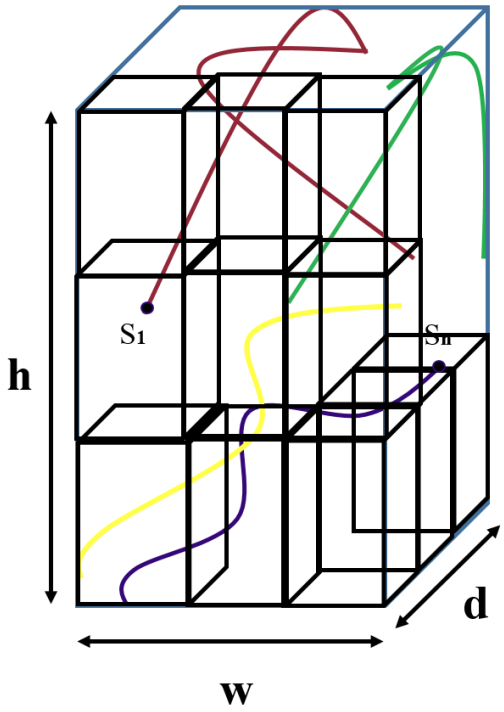

FIGURE 3.16 – Illustration de la construction d'un histogramme de zoning 3D flou.

$$m_{pqr} = \sum_{i=1}^{n} (x_i - \mu_x)^p (y_i - \mu_y)^q (z_i - \mu_z)^r \tag{3.27}$$

Les moments sont ensuite normalisés, pour garantir une indépendance par rapport à l'échelle :

$$\nu_{pqr} = \frac{m_{pqr}}{m_{000}^{\gamma}}, \qquad \gamma = 1 + \frac{p+q+r}{3} \tag{3.28}$$

Les trois descripteurs invariants sont ensuite calculés comme indiqué dans [SH80] :

$$\begin{aligned}
\mathbf{f_{85}} &= \nu_{200} + \nu_{020} + \nu_{002}, \\
\mathbf{f_{86}} &= \nu_{200}\nu_{020} + \nu_{200}\nu_{002} + \nu_{020}\nu_{002} - \nu_{110}^2 - \nu_{101}^2 - \nu_{011}^2, \\
\mathbf{f_{87}} &= \nu_{200}\nu_{020}\nu_{002} + 2\nu_{110}\nu_{101}\nu_{011} \\
&\quad - \nu_{002}\nu_{110}^2 - \nu_{020}\nu_{101}^2 - \nu_{200}\nu_{011}^2
\end{aligned} \tag{3.29}$$

Descripteurs basés sur l'enveloppe convexe : Les deux derniers descripteurs capturent la forme 3D du pattern en considérant son enveloppe convexe. En fait, l'enveloppe convexe H du pattern S est d'abord calculée au moyen de l'algorithme quickhull [BDH96] pour ensuite déduire son volume V_H. Les deux descripteurs associés sont le volume de l'enveloppe

convexe normalisé par le volume de la boîte englobante, et la compacité :

$$\mathbf{f_{88}} = \frac{V_H}{\mathbf{w} * \mathbf{h} * \mathbf{d}}, \qquad \mathbf{f_{89}} = \frac{L^3}{V_H} \tag{3.30}$$

Comme pour les descripteurs de tracés manuscrits, **HIF3D** ne capturent que les propriétés de forme et les informations spatiales relatives aux trajectoires produites par une action. Pour extraire l'information de dépendance temporelle au sein du pattern, nous avons repris la même stratégie que pour la représentation **3DMM** où l'on opère une division temporelle de la séquence à plusieurs niveaux. Nous avons considéré deux niveaux temporels, comme pour l'approche **3DMM**, ce qui porte la dimension finale de la représentation à 356 (89 * 1 + 89 * 3).

Pareillement, nous nous basons pour l'étape de classification sur des machines à vecteurs de support (SVM) avec un noyau polynomial et des paramètres optimisés expérimentalement au moyen d'une grille de recherche sur un ensemble de validation. Les deux représentations seront donc directement comparables pour mettre en avant la contribution des descripteurs **HIF3D** et ceux utilisés dans **3DMM**.

Mais à l'opposé de l'approche **3DMM**, nous n'effectuons aucune étape de sélection de descripteurs afin d'évaluer la performance d'une même représentation pour toutes les bases d'actions et suivant tous les protocoles de test.

3.4 Résultats expérimentaux et discussion

Dans cette section, nous présentons les évaluations des deux représentations proposées **3DMM** et **HIF3D**. Pour ce faire, nous avons sélectionné trois bases d'actions 3D présegmentées squelettiques qui servent souvent de benchmarks dans la communauté : M2S-dataset [SKBM13], UTKinect-Action [XCA12] et HDM05 [MRC+07]. Dans ce qui suit, nous considérons d'abord M2S-dataset qui est une base interne à notre laboratoire. Nous présentons ensuite les résultats sur les benchmarks internationaux UTKinect-Action puis HDM05.

3.4.1 Base de données M2S-dataset

La base de données M2S-dataset comporte des actions haut du corps où seuls les deux bras sont impliqués [SKBM13]. Certaines de ces actions sont très ressemblantes car elles partagent plusieurs caractéristiques géométriques et de formes. Les propriétés des actions qui composent cette base sont synthétisées dans la Table 3.1.

En particulier, cette base comprend 15 classes d'actions dont applaudir, donner une claque avec la paume, donner une claque avec le dos de la main, poser ses mains sur les

	Label des classes	Nombre de séquences par classe	Nombre de sujets
	Applaudir (Applause)	70	-
	Croiser les bras (Cross arms)	68	-
	Prendre milieu (Grasp chest)	68	-
	Prendre haut (Grasp high)	70	-
	Prendre bas (Grasp hip)	68	-
	Mains sur les hanches (Hand hip)	70	-
	Mains dans les poches (Hand pocket)	68	-
	Saluer au niveau de la tête (Hello head)	70	-
	Saluer haut (Hello high)	70	-
	Coup de poing (Punch)	68	-
	Coup de poing remontant (Uppercut)	72	-
	Claque revers (Slap back)	70	-
	Claque paume (Slap palm)	72	-
	Jeter (Throw)	70	-
	Toucher le menton (Touch chin)	68	-
Total	15 classes	1042 séquences	10 sujets

TABLE 3.1 – Tableau récapitulatif des propriétés de la base M2S-dataset en termes de nature des actions, nombre de classes d'actions, nombre de séquences et nombre total des sujets.

hanches, etc. Chaque classe d'action a été réalisée par 10 sujets, au moins cinq fois de chaque côté. Il a été demandé aux sujets d'inclure une variabilité élevée (en termes de vitesse, de positionnement et d'amplitude). Les données fournies contiennent des informations complètes sur le corps mais, pour une comparaison équitable avec les travaux précédents [Sor12], seules les données des deux bras ont été utilisées dans nos expérimentations.

Nous avons suivi le même protocole proposé par [Sor12] consistant en une validation croisée avec 10 folds. Les auteurs ont utilisé un modèle séquentiel où chaque classe d'action est codée comme un HMM à sept états. Nous rapportons dans la Table 3.2 les résultats de l'approche **3DMM** et de **HIF3D** pour le cas où la temporalité n'est pas incluse dans la représentation, c'est-à-dire les descripteurs sont uniquement extraits sur l'intégralité de la séquence, ce que l'on note par niveau = 1 dans la Table. Nous rapportons aussi les résultats des deux approches lorsque la temporalité est considérée en ajoutant aux descripteurs du premier niveau les descripteurs extraits sur les sous-séquences, ce que l'on note par niveau = 2. En outre, pour les raisons évoquées à la section 3.2.2.3 nous opérons une sélection dans l'approche **3DMM**. Afin d'évaluer l'impact de cette sélection, nous présentons les résultats obtenus par l'approche **3DMM** avec l'intégralité des descripteurs mais aussi avec un sous-ensemble de descripteurs sélectionnés sur un ensemble

Approches	# Descripteurs	Taux de reco. (%)
HMM [Sor12]	-	93.60
3DMM + SVM + Niveau = 1	147 (Total)	**92.99**
3DMM + SVM + Niveau = 1	70	**94.24**
3DMM + SVM + Niveau = 2	588 (Total)	**95.20**
3DMM + SVM + Niveau = 2	280	**95.29**
HIF3D + SVM + Niveau = 1	89 (Total)	**94.34**
HIF3D + SVM + Niveau = 2	356 (Total)	**95.77**

TABLE 3.2 – Reconnaissance, M2S : Comparaison entre les approches **3DMM** et **HIF3D** avec l'approche de [Sor12] sur la base d'actions M2S-dataset en utilisant un classifieur SVM.

de validation.

Les résultats rapportés dans la Table 3.2 permettent d'affirmer que les deux approches que nous proposons sont globalement performantes sur cette base d'actions. En particulier, il est possible de tirer quatre conclusions à partir de ces résultats. D'abord le fait d'inclure la temporalité en extrayant les mêmes descripteurs sur des sous-séquences améliore les performances de reconnaissance. Ce constat est valable aussi bien pour l'approche **3DMM**, qui réalise un score de **94.24%** sans inclure de temporalité alors qu'elle obtient un score de **95.29%** pour une hiérarchie à deux niveaux, que pour la représentation à base des descripteurs **HIF3D**, qui lui permet de passer de **94.34%** à un score de **95.77%** en incluant la temporalité.

Deuxièmement, puisque nous avons appliqué les mêmes opérations de prétraitement amorphologique que celles déployées dans l'approche concurrente de [Sor12], nous pouvons dire que l'amélioration du score est spécifiquement due aux descripteurs utilisés dans nos deux représentations.

Troisièmement, suite à des évaluations sur un ensemble de validation, nous avons sélectionné les meilleurs descripteurs (70 pour niveau = 1 et 280 pour niveau = 2) suivant l'approche **3DMM**. Au vu de l'amélioration des résultats, nous pouvons conclure de l'existence d'une certaine redondance due notamment à l'extraction des mêmes descripteurs sur des trajectoires 3D projetées sur des plans.

Enfin, bien que l'approche **3DMM** soit une approche prometteuse, il est clair que le jeu de descripteurs **HIF3D** nouvellement conçu permet de réaliser d'aussi bons résultats que

ceux de l'approche **3DMM** sans devoir opérer de sélection à l'opposé de cette dernière.

Les résultats que nous obtenons sont d'autant plus intéressants que la base de données M2S-dataset contient des gestes difficilement différentiables, ce qui implique une forte similarité entre les classes. Nous illustrons deux aspects de cette complexité dans la Figure 3.17 et la Figure 3.18. Sur la Figure 3.17, nous donnons des exemples de trois classes qui présentent la même propriété temporelle et de très faibles variabilités inter-classes, ce qui les rend difficilement distinguables. Sur la Figure 3.18, nous présentons les deux actions de "donner une claque avec la paume" et de "donner une claque avec le revers de la main". Ces actions sont plus facilement séparables sur le plan temporel vu qu'elles produisent des formes globales identiques mais sont temporellement opposées. Ceci montre l'intérêt d'inclure l'information temporelle dans nos représentations.

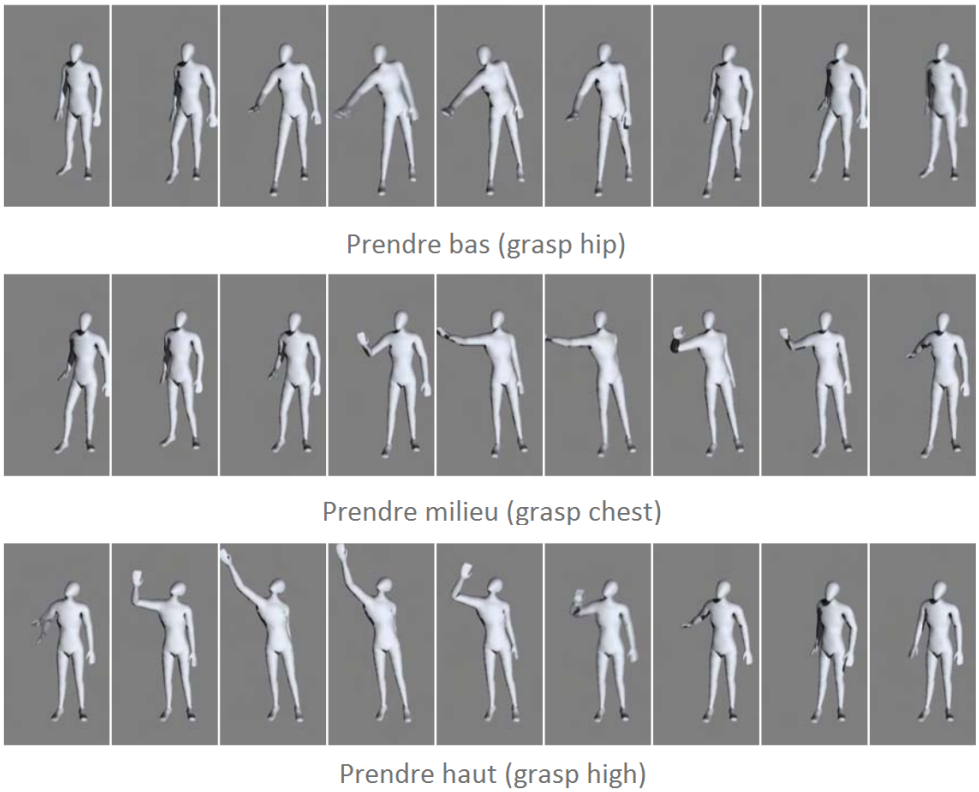

FIGURE 3.17 – M2S-dataset : Illustration des trois classes d'actions "Prendre bas", "Prendre milieu" et "Prendre haut" qui présentent des propriétés spatiales très similaires.

3.4.2 Base de données UTKinect-Action

Pour évaluer nos deux représentations sur des données plus bruitées, nous effectuons deux autres expérimentations sur une autre base, UTKinect-Action. Cette base a été

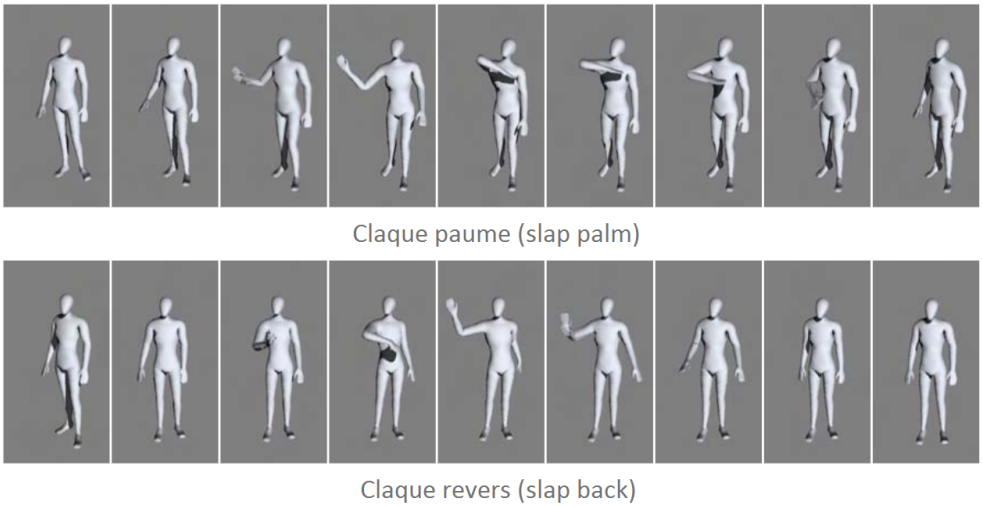

FIGURE 3.18 – M2S-dataset : Illustration des deux classes d'actions "Claque paume" et "Claque revers" qui produisent le même pattern mais sont temporellement symétriques.

collectée en utilisant un capteur Kinect stationnaire [XCA12]. Elle se compose de 10 actions effectuées par 10 sujets différents. Chaque sujet a effectué toutes les actions deux fois. Au total, il y a 200 séquences d'actions. Les positions 3D de 20 articulations sont fournies. Il s'agit d'une base bruitée (capturée avec une Kinect) mais tous les sujets effectuent leurs actions en restant toujours devant la caméra et à une même distance. Nous résumons les propriétés de cette base dans la Table 3.3.

	Label des classes	Nombre de séquences par classe	Nombre de sujets
	Marcher (Walk)	20	10
	S'asseoir (Sit down)	20	10
	Se lever (Stand up)	20	10
	Ramasser (Pick up)	20	10
	Porter (Carry)	20	10
	Jeter (Throw)	20	10
	Pousser (Push)	20	10
	Tirer (Pull)	20	10
	Agiter les mains (Wave hands)	20	10
	Battre des mains (Clap hands)	20	10
Total	10 classes	200 séquences	10 sujets

TABLE 3.3 – Tableau récapitulatif des propriétés de la base UTKinect-Action en termes de nature des actions, nombre de classes d'actions, nombre de séquences et nombre total des sujets.

Nous avons d'abord évalué nos représentations selon le protocole LOSeqO (Leave-One-Sequence-Out) proposé par [XCA12]. Il consiste à laisser une seule séquence pour le test tandis que les 199 autres séquences sont utilisées pour l'apprentissage. Suivant ce protocole, le sujet ayant effectué la séquence de test figure aussi parmi les sujets qui composent l'ensemble d'apprentissage. Les résultats de l'expérience sont présentés dans la Table 3.4.

Sur la base de ces scores, il est possible de confirmer que nos représentations permettent de réaliser de meilleures performances que les approches de l'état de l'art. De plus, nous relevons que les performances des deux approches proposées sont comparables (94.00% avec **HIF3D** et 93.50% avec **3DMM** sans sélection), avec un léger avantage pour l'approche **3DMM** lorsqu'une sélection des meilleurs descripteurs est opérée (96.00%). Ceci atteste globalement du bien-fondé du transfert du savoir-faire 2D vers la modélisation de trajectoires 3D.

Il est important de relever que cette base comporte des actions plus facilement distinguables que ceux de la base précédente (M2S-dataset) et que de ce fait l'approche **3DMM** qui extrait des descripteurs après projection et en sélectionne les meilleurs suffit

Approche	Walk	Sit	Stand	Pick	Carry	Throw	Push	Pull	Wave	Clap	**Taux de reco. (%)**
LTI + HMM [PLCSC14]	**63.16**	100	100	100	83.33	**61.11**	90	100	85	85	86.76
Grassmann + SVM [SWDS15]	100	80	100	100	100	**60**	**65**	85	100	95	88.50
HOJ3D + HMM [XCA12]	96.5	91.5	93.5	97.5	97.5	**59**	81.5	92.5	100	100	90.95
DS-SRC + Nearest neighbour [TKEF14]	90	100	95	85	100	**75**	90	95	100	80	91.00
STFC + SVM [DLCZ15]	90	95	95	100	**65**	90	95	100	100	85	91.50
HSOM + VMM [DLCJ16]	-	-	-	-	-	-	-	-	-	-	94.50
3DMM + SVM + Niveau = 1 + 147 descripteurs (Total)	85	100	100	75	**85**	**90**	95	100	100	95	**92.50**
3DMM + SVM + Niveau = 1 + 90 descripteurs	85	100	100	95	**90**	**100**	95	100	100	95	**96.00**
3DMM + SVM + Niveau = 2 + 588 descripteurs (Total)	90	95	100	90	**85**	**90**	100	100	100	85	**93.50**
3DMM + SVM + Niveau = 2 + 360 descripteurs	85	100	100	95	**90**	**100**	95	100	100	95	**96.00**
HIF3D + SVM + Niveau = 1 + 89 descripteurs (Total)	85	100	100	80	85	85	80	100	100	90	**90.50**
HIF3D + SVM + Niveau = 2 + 356 descripteurs (Total)	95	100	100	85	90	95	90	100	100	85	**94.00**

TABLE 3.4 – Reconnaissance, UTKinect : Résultats des deux représentations **3DMM** et **HIF3D** et ceux approches précédentes sur la base de données UTKinect-Action selon le protocole de LOSeqO. Les taux de reconnaissance pour chaque classe ainsi que le taux global (%) sont donnés.

pour atteindre des scores importants. De plus, nous avons tenu à utiliser l'intégralité des descripteurs **HIF3D** et éviter de faire une sélection. En effet, en comparant le nombre de descripteurs sélectionnés dans la Tables 3.2 et 3.4, il apparaît clairement que ce nombre varie d'une base à une autre et que de ce fait la représentation **3DMM** n'est pas généralisable, ce qui est problématique. Par ailleurs, il est possible de constater qu'avec les mêmes descripteurs **HIF3D**, nous réussissons à réaliser de très bons scores sur cette deuxième base aussi.

Néanmoins, ces résultats permettent de mettre en évidence une possibilité d'amélioration importante pour le jeu de descripteurs développé **HIF3D**. Il est possible de dire en effet que certains descripteurs sont inutiles, voire même nuisibles aux performances de reconnaissance. Il serait alors intéressant d'analyser l'impact de chacun des descripteurs ainsi étendus pour éventuellement constituer un nouveau jeu de descripteurs plus performant mais sans devoir à chaque fois effectuer une sélection. Ce point fait partie des perspectives de cette thèse.

Une autre expérimentation est menée selon le protocole proposé par [DLCZ15]. Il s'agit d'entraîner le modèle sur les données issues de 5 sujets (50%) et de tester sur les données restantes. Cette évaluation est répétée pour toutes les combinaisons possibles de sujets. Ceci est plus complexe que le protocole LOSeqO, où la totalité des sujets sont vus lors de l'apprentissage. Les taux de reconnaissance sont présentés dans la Table 3.5.

Les résultats rapportés dans la Table 3.5 concernant cette deuxième expérimentation

Approche	**# Descripteurs**	**Taux de reconnaissance (%)**
STFC [DLCZ15]	-	85.00
Joint features [ZCG13]	-	87.90
3DMM + SVM + Niveau = 1	147 (Total)	**86.07**
3DMM + SVM + Niveau = 1	100	**90.46**
3DMM + SVM + Niveau = 2	588 (Total)	**89.18**
3DMM + SVM + Niveau = 2	400	**91.51**
HIF3D + SVM + Niveau = 1	89 (Total)	**85.33**
HIF3D + SVM + Niveau = 2	356 (Total)	**90.96**

TABLE 3.5 – Reconnaissance, UTKinect : Comparaison des résultats des représentations **3DMM** et **HIF3D** avec ceux obtenus par deux approches de l'état de l'art sur la base de données UTKinect-Action, selon le protocole de combinaison de sujets proposé dans [DLCZ15].

confirment globalement ceux déjà obtenus lors de la première expérimentation (Table 3.4). En effet, il est possible de noter que les deux représentations améliorent les scores de reconnaissance en comparaison aux approches de l'état de l'art. Il est aussi possible d'en déduire que nos représentations permettent de mieux adresser la variabilité morphologique en comparaison aux autres approches, vu que le protocole suivi dans cette deuxième expérimentation (répartition de données par sujets) accentue cette variabilité. Ceci justifie notamment la baisse des résultats par rapport à ceux obtenus lors de la première expérimentation sur la base UTKinect-Action (Table 3.4), où les données d'apprentissage comprennent des échantillons effectués par les sujets de test.

3.4.3 Base de données HDM05

La dernière expérimentation que nous présentons est menée sur une partie de la base HDM05. Il s'agit d'une base de données squelettiques qui a été collectée via un système de capture opto-électronique [MRC+07]. Cette base contient une centaine de classes de mouvement, dont divers mouvements de marche et de course, des mouvements de rotation et de saisis, des mouvements accroupis, etc. Chaque classe de mouvement contient 10 à 50 occurrences différentes, couvrant un large spectre d'actions humaines corps-complet. Les propriétés du sous-ensemble considéré pour notre évaluation sont résumées dans la Table 3.6.

	Label des classes	**Nombre de frames**	**Nombre de séquences**	**# séquences par sujet**				
				bd	bk	dg	mm	tr
1	Déposer par terre (Deposit floor)	11,623	32	6	6	6	8	6
2	Courir sur place (Elbow to knee)	5756	13	3	3	3	1	3
3	Grab high	7506	29	4	6	6	6	7
4	Sauter pieds joints (Hop both legs)	2327	12	4	3	3	1	1
5	Courir en demi-cercle (Jog)	4142	17	2	5	3	3	4
6	Coup de pied vers l'avant (Kick forward)	6225	29	6	6	6	6	5
7	S'allonger par terre (Lie down floor)	13,100	20	4	6	4	4	2
8	Tournez les bras en arrière (Rotate both arms backward)	1742	17	4	4	3	3	3
9	Marcher (Sneak)	3480	16	3	3	4	3	3
10	S'accroupir (Squat)	10,035	50	10	12	12	4	12
11	Lancer un ballon (Throw basketball)	5710	14	3	3	3	2	3
Total	11 classes	71,646	249	49	57	53	41	49

TABLE 3.6 – Tableau récapitulatif des propriétés de la base HDM05 en termes de nature des actions, nombre de classes d'actions, nombre de frames par classe, nombre de séquences par classe et par sujet et nombre total des sujets.

Plusieurs études ont déjà été menées sur la base de données HDM05. Pour notre évaluation, nous adoptons la configuration expérimentale de [OCK+14] qui suggère un sous-ensemble de 11 actions. Les actions sont effectuées par 5 sujets, tandis que chaque

sujet effectue chaque action une ou deux fois ; ce qui donne un total de 249 séquences. Comme pour [OCK+14], nous adoptons un protocole suivant lequel les données de trois sujets (référencés par bd, mm et tr) sont utilisées pour l'étape d'apprentissage et celles de deux autres sujets (référencés par bk et dg) pour le test. Ce protocole résulte sur une configuration de 139 échantillons pour l'entraînement et 110 pour le test. Les résultats de nos deux représentations **3DMM** et **HIF3D** sont rapportés dans la Table 3.7. Dans cette table nous avons aussi rapporté les résultats de huit approches de l'état de l'art traitant de la reconnaissance d'actions squelettiques.

Plusieurs conclusions peuvent être tirées des résultats rapportés dans la Table 3.7. Pour ce qui est de la représentation **3DMM**, correspondant à une modélisation d'actions 3D par projection dans un espace de représentation 2D, il apparaît d'abord que l'ajout d'informations suivant un découpage temporel permet d'améliorer sensiblement les performances de nos deux approches. En effet, la représentation **3DMM** à deux niveaux temporels sans sélection réalise un taux de reconnaissance de **93.20%** alors que ce taux chute à **90.99%** en utilisant le même classifieur, la totalité des descripteurs (147) mais sans découpage temporel (Niveau = 1). Ceci confirme donc l'intérêt de découper une sé-

Approche & Année	**# Descripteurs**	**Taux de reco. (%)**
SMIJ + SVM, 2014 [OCK+14]	-	84.47
MIJA/MIRM + LCSS, 2015 [PDLM15]	-	85.23
SMIJ + Nearest neighbour, 2014 [OCK+14]	-	91.53
LDS + SVM, 2013 [COK+13]	-	91.74
Skeletal Quads + SVM, 2014 [ESH14]	9360	93.89
Cov3DJ + SVM, 2013 [HTGES13]	43710	95.41
BIPOD + SVM, 2015 [ZP15]	-	96.70
HOD + SVM, 2013 [GTHES13]	1116	97.27
3DMM + SVM + Niveau = 1	147 (Total)	**90.99**
3DMM + SVM + Niveau = 1	100	**91.74**
3DMM + SVM + Niveau = 2	588 (Total)	**93.20**
3DMM + SVM + Niveau = 2	400	**94.49**
HIF3D + SVM + Niveau = 1	89 (Total)	**90.83**
HIF3D + SVM + Niveau = 2	356 (Total)	**98.17**

TABLE 3.7 – Reconnaissance, HDM05 : Résultats expérimentaux des deux représentations **3DMM** et **HIF3D** suivant le protocole proposé par [OCK+14] sur la base HDM05. Nos deux représentations ont été évaluées avec le classifieur SVM en opérant ou non un découpage temporel (Niveau = 2 et Niveau = 1, respectivement).

quence en plusieurs sous-séquences temporelles pour capter les dépendances temporelles entre les trajectoires d'une même action. Ces résultats sont néanmoins meilleurs si la sélection est appliquée et les scores sont égaux à **91.74%** et **94.49%** pour des niveaux de temporalité égaux, respectivement, à 1 et 2.

De plus, en comparaison à la plupart des approches de l'état de l'art, la représentation "naïve" **3DMM** réalise des performances intéressantes et arrive même à surpasser certaines approches beaucoup plus complexes telles que la représentation SMIJ [OCK+14] ou la représentation de Skeletal Quads [ESH14].

Enfin, malgré le fait que l'approche **3DMM** soit une première manière simple de tirer profit du savoir-faire 2D pour modéliser des actions 3D, le score élevé qu'elle réalise témoigne déjà du bien-fondé de mener ce transfert. En effet, les résultats obtenus confirment le grand potentiel de modélisation des descripteurs retrouvés dans la littérature de la 2D, en particulier HBF49 [DA13].

S'agissant maintenant de la représentation **HIF3D**, résultant du transfert des descripteurs 2D dans l'espace de représentation 3D, il est possible de relever que cette représentation réalise un nouveau score, à savoir **98.17%** et surpasse ainsi toutes les autres approches évaluées suivant ce protocole expérimental. En effet, cette nouvelle représentation est plus performante que la représentation **3DMM**. Ceci confirme notre intuition qu'il est plus intéressant d'étendre le savoir-faire vers l'espace 3D que de ramener le problème dans l'espace 2D, notamment à cause de la perte d'information de corrélation engendrée par la projection.

La supériorité de **HIF3D** par rapport à la représentation **3DMM** s'est beaucoup plus révélée sur la base HDM05, car cette base est composée d'actions plus complexes (fortes similarités) et que pour certaines le sujet se déplace librement dans une salle et ne reste pas fixe devant une caméra comme c'est le cas pour les deux bases évaluées précédemment. Cette complexification de la tâche de reconnaissance a permis de mettre plus en avant le potentiel de modélisation supérieur du nouveau jeu de descripteurs **HIF3D** qui considère la trajectoire en 3D au lieu de modéliser les simples projections. Des illustrations de certaines actions constituant la base HDM05 sont données dans la Figure 3.19.

En outre, la représentation **HIF3D** est beaucoup plus simple que la plupart des approches précédentes, et ne nécessite pas toutes les données articulaires. Enfin, la représentation **HIF3D** est d'autant plus intéressante qu'elle n'est composée que d'un faible nombre de descripteurs. En effet, notre représentation est composée de deux niveaux avec 89 descripteurs seulement par partition. Au contraire, l'un des meilleurs résultats précédent [HTGES13] a été atteint au moyen d'une hiérarchie à trois niveaux temporels avec 1830 descripteurs pour chaque partition temporelle. Ceci est notamment important pour des applications interactives nécessitant une reconnaissance en temps réel d'actions 3D,

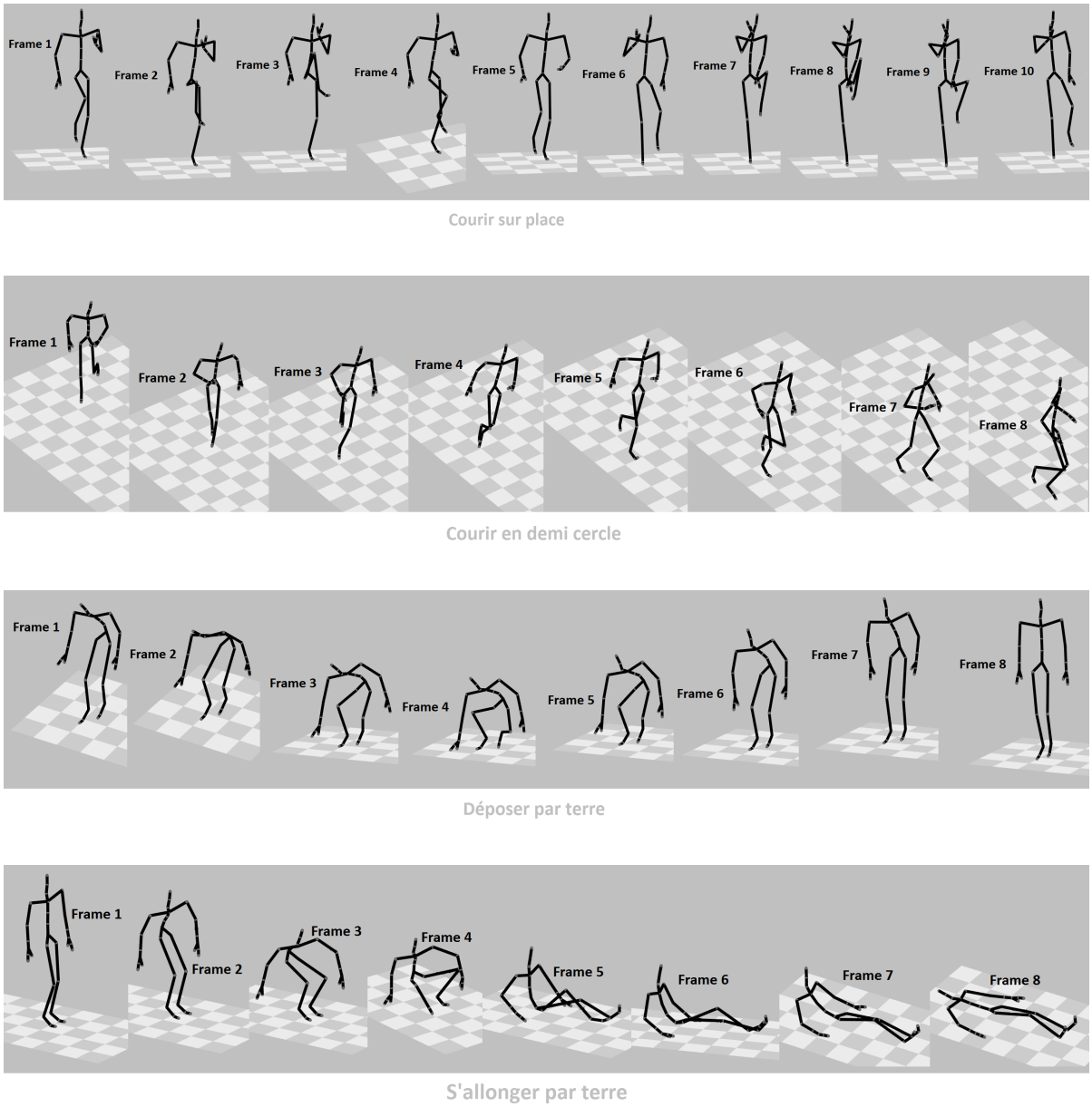

FIGURE 3.19 – HDM05 : Illustration de quatre classes d'actions de la base HDM05 qui présentent de fortes similitudes dont d'une part "courir sur place" et "courir en demi-cercle" et d'autre part "déposer par terre" et "s'allonger".

une problématique que nous nous proposons d'aborder dans le chapitre 4. Par conséquent, l'efficacité de notre approche est confortée par sa simplicité par rapport à d'autres propositions.

Au vu des scores obtenus, il est ainsi possible de conclure de la pertinence des représentations proposées, notamment le nouveau jeu de descripteurs **HIF3D**, mais plus globalement du bien-fondé de l'idée de s'inspirer des approches de représentation de tra-

jectoires 2D pour concevoir des descripteurs robustes pour des trajectoires 3D.

3.5 Conclusion

Nous avons présenté dans ce chapitre une nouvelle piste de recherche consistant à s'inspirer du savoir-faire des représentations de tracés manuscrits 2D pour modéliser des actions squelettiques appartenant à un large panel d'applications indépendamment du nombre de trajectoires induites. Il peut s'agir donc des actions effectuées par les bras et les jambes, les bras seuls, les jambes seules, la main seule, etc. Cette proposition part de l'observation suivante : les patterns produits par un mouvement humain, en particulier ceux traduisant des tracés 2D et des actions 3D, partagent plusieurs propriétés importantes relatives aux informations spatio-temporelles induites par les trajectoires qu'elles sous-tendent. C'est pourquoi nous avons émis l'hypothèse que les deux problèmes de reconnaissance pourraient être traités de manière similaire.

Pour ce faire, nous avons d'abord identifié trois difficultés majeures à considérer lors de la conception d'une nouvelle représentation d'actions 3D pré-segmentées. Il s'agit d'adresser la variabilité morphologique entre différents sujets, de prendre en compte la corrélation spatiale entre différentes trajectoires articulaires et enfin de considérer le séquencement temporel au sein d'une action. A la suite de cela, nous avons mis en place et évalué deux représentations d'actions 3D squelettiques. Chacune de ces représentations correspond à une approche différente inspirée du savoir-faire 2D, tout en répondant explicitement aux trois difficultés identifiées.

La première représentation, appelée **3DMM** pour *3D Multistroke Mapping*, correspond au transfert de la problématique de reconnaissance d'actions 3D dans un espace de modélisation 2D. En particulier, nous avons projeté sur 3 plans orthogonaux les trajectoires issues des vecteurs amorphologiques retenus. Ensuite, des descripteurs 2D, à savoir HBF49 [DA13], ont été extraits sur les projections obtenues dans chacun des plans. Ces trajectoires projetées sont regroupées sous forme d'un tracé dit multistrokes permettant de capturer les dépendances spatiales entre les trajectoires. Enfin, pour inclure l'information du séquencement temporel, nous avons extrait les mêmes descripteurs suivant un découpage chevauchant temporel de la séquence.

La deuxième représentation, appelée **HIF3D** pour *Handwriting-Inspired Features for 3D skeleton-based action recognition*, est un nouveau jeu de descripteurs dédié aux actions 3D squelettiques. Il s'agit d'un transfert du savoir-faire 2D à l'espace de représentation 3D. **HIF3D** est donc une extension aux actions 3D squelettiques du jeu de descripteurs 2D HBF49 [DA13], que nous avons déjà utilisé dans l'approche **3DMM**. A la différence de ce qui a été fait pour l'approche **3DMM**, les trajectoires amorphologiques formées sont rapportées dans le repère 3D ayant pour origine le centre articulaire de la hanche. Il en résulte un pattern composé de strokes 3D, c'est-à-dire un pseudo-symbole multistrokes 3D,

à partir duquel les descripteurs **HIF3D** sont extraits. Le jeu **HIF3D** proposé comporte 89 descripteurs, dont 41 descripteurs étendus et 48 descripteurs nouveaux. Les 41 descripteurs sont issus d'une extension directe (c'est-à-dire même formulation mathématique que pour la 2D) alors que les 48 restants sont des descripteurs qui extraient la même information que leurs équivalents 2D mais sont formulés différemment.

Les experimentations ont été menées sur trois bases de données d'actions 3D, à savoir M2S-dataset [SKBM13], UTKinect-Action [XCA12] et HDM05 [MRC+07]. Les résultats obtenus suivant différents protocoles dépassent les résultats de l'état de l'art. Ceci est d'autant plus significatif que les approches précédentes sont plus complexes et utilisent des espaces de représentation beaucoup plus grands.

Enfin, il est important de préciser que notre objectif était d'atteindre les résultats de l'état de l'art tout en ayant un nombre réduit de descripteurs. Ceci nous ouvre des perspectives prometteuses en termes de compacité des représentations (temps de calcul réduit) pour aborder des problèmes plus complexes que sont la reconnaissance d'actions dans un flot non segmenté ainsi que la reconnaissance précoce comme nous allons le voir dans le chapitre suivant.

Chapitre 4

Détection en-ligne d'actions 3D dans un flot non segmenté

4.1 Introduction

La détection en-ligne d'actions (**OAD** pour *Online Action Detection*) est devenue récemment une problématique très importante dans le domaine de la vision par ordinateur et de l'apprentissage automatique [VFML17, LKK16]. A la différence de la reconnaissance d'actions pré-segmentées, qui se résume à l'étiquetage d'une **seule action** de manière hors ligne et après que cette action soit complètement achevée, la OAD opère sur un flot de frames continu (non segmenté) pour identifier en temps réel des actions dont **ni le début, ni la fin, ni le nombre ne sont connus par avance**. Ceci offre une compréhension en temps réel de ce que le sujet est en train de faire pour permettre le renvoi d'un feedback approprié. Afin de clarifier la différence entre la reconnaissance d'actions pré-segmentées et la OAD, nous illustrons dans la Figure 4.1 d'une part, le type de pattern reçu en entrée par une approche OAD et d'autre part, celui reçu par une approche de reconnaissance d'actions pré-segmentées.

Certains travaux [HYWDLT14, EMS16, BAM17] proposent des approches permettant une identification des actions le plus tôt possible, c'est-à-dire avec très peu de frames observées. L'intérêt de ces approches est d'améliorer la réactivité d'un système d'interaction. Dans les premiers travaux de la littérature [HYWDLT14, EMS16], cette tâche est appelée **reconnaissance précoce** car les instants de début et de fin étaient connus par avance. Dans ces travaux, il n'était question que d'identifier la classe d'action avec le moins de frames possible et avant la fin du geste. Les travaux récents [BAM17, BAM16, BMA14, BAM13] s'intéressent à ce qu'il faudrait appeler désormais la **détection précoce**. Cette dernière tâche est opérée sur des flots continus et en temps

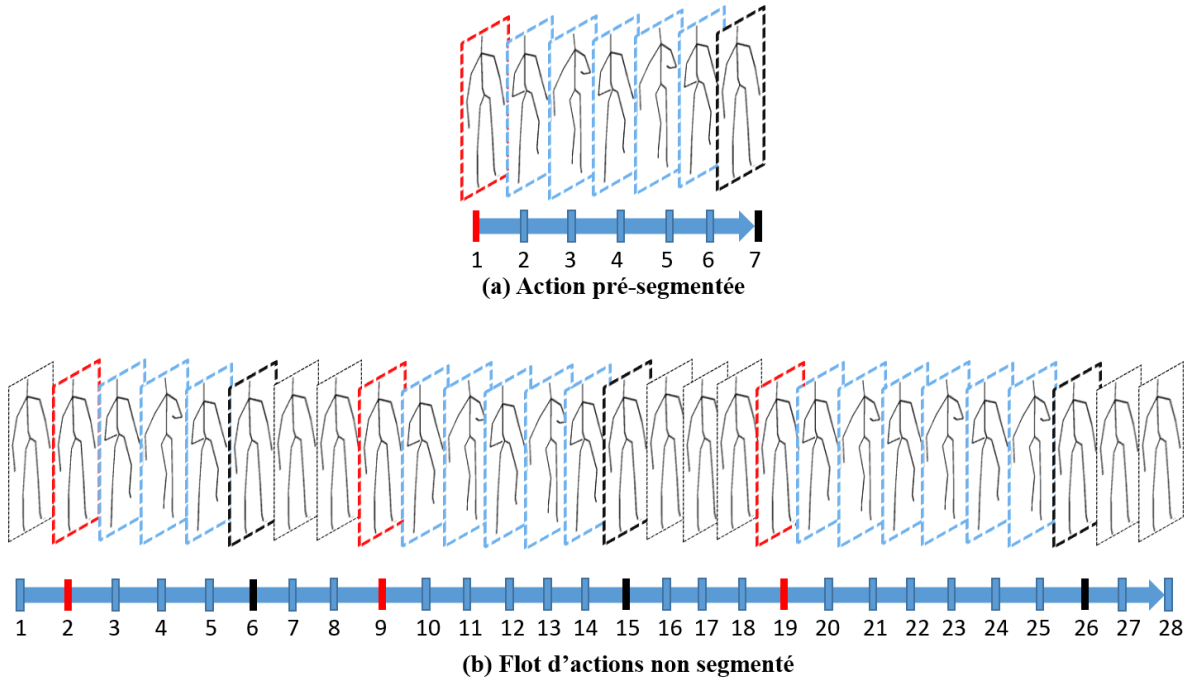

FIGURE 4.1 – Illustration **(a)** d'une action pré-segmentée qui correspond au pattern reçu en entrée par une approche de reconnaissance d'actions pré-segmentées et **(b)** d'un flot d'actions non segmenté utilisé pour une approche OAD. Dans cet exemple, il est constitué de trois performances.

réel, dans les mêmes conditions que la OAD.

Dans ce chapitre, nous proposons d'abord d'adresser la problématique de la OAD. Nous présenterons ensuite nos travaux sur la détection précoce qui constitue un des défis les plus complexes à aborder.

4.2 Détection en-ligne d'actions 3D : OAD

En suivant la même démarche que celle adoptée pour adresser la problématique de reconnaissance d'actions pré-segmentées (chapitre 3), nous proposons d'abord d'identifier et de discuter les difficultés majeures pour la tâche de la OAD avant de présenter les solutions proposées. En particulier, notre approche OAD opère suivant un nouveau concept de segmentation. Ce concept considère le flot en entrée comme un flot spatial et non un flot temporel, en se basant notamment sur le déplacement curviligne des articulations squelettiques. Pour cette raison, nous dénommons cette approche **CuDi3D** pour *Curvilinear Displacement based approach for online 3D action detection.*

4.2.1 Difficultés relevées pour la détection en-ligne d'actions 3D

En plus des difficultés déjà évoquées pour représenter des actions pré-segmentées, nous avons relevé trois autres difficultés majeures pour la tâche de la OAD, à savoir la *variabilité temporelle*, la *variabilité spatiale inter-classes* et la *variabilité spatiale intra-classe*.

4.2.1.1 Comment adresser la variabilité temporelle ?

Le premier problème soulevé est la *variabilité temporelle* qui peut avoir lieu lorsque des sujets exécutent les mêmes actions avec des vitesses différentes et/ou avec des pauses insérées avant, pendant ou après l'action. Cette variabilité rend difficile le fait de prédire la durée totale que prendra une action qui serait en cours d'exécution.

Or, la plupart des approches OAD de la littérature analysent le flot d'entrée au moyen de fenêtres temporelles glissantes de tailles pré-déterminées (égales à un nombre déterminé au préalable de frames). Ce faisant, ces approches OAD extraient en phase de test des descripteurs sur des segments temporellement fixes, sans considérer la cohérence temporelle avec ce qui a été appris par le classifieur.

Si l'on ne considère pas cette variabilité temporelle, il est impossible d'avoir des approches robustes à même d'être exploitées dans un cas d'utilisation réel. Dans ces travaux de thèse, nous avons explicitement considéré cette difficulté lors de la conception de notre approche OAD.

4.2.1.2 Comment adresser la variabilité spatiale inter-classes ?

La seconde difficulté identifiée, appelée *variabilité spatiale inter-classes*, est due au fait que lorsqu'un sujet effectue différentes classes d'actions, il est probable que les longueurs des déplacements associés à chaque performance soient différentes. Pour illustrer cette variabilité, nous schématisons dans la Figure 4.2a les déplacements squelettiques qui résultent de la performance de deux classes d'actions Ci et Cj. Il est possible de noter que le déplacement associé à la classe Ci est inférieur à celui induit par la classe Cj. Dans un flot non segmenté où plusieurs actions peuvent s'enchaîner, cette variabilité spatiale inter-classes rend difficile la détermination des segments de mouvements à considérer pour la détection d'une potentielle action. Cette difficulté est accrue lorsqu'il existe des classes d'actions qui partagent des débuts communs comme c'est le cas pour les deux classes Ci et Cj illustrées dans la Figure 4.2a. Si cette variabilité n'est pas considérée, une décision (erronée) de détection peut alors être émise bien avant que l'action en cours ne soit identifiable.

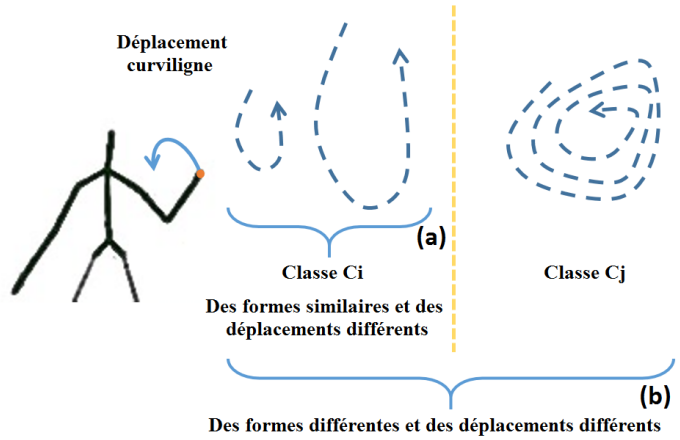

FIGURE 4.2 – Illustration via la trajectoire d'une seule articulation **(a)** de la *variabilité spatiale inter-classes* entre deux classes Ci et Cj et **(b)** de la *variabilité spatiale intra-classe*.

4.2.1.3 Comment adresser la variabilité spatiale intra-classe ?

La troisième et dernière difficulté relevée est la *variabilité spatiale intra-classe*. Cette variabilité représente une propriété intrinsèque du mouvement humain. Elle conduit souvent au fait que des instances, appartenant pourtant à la même classe d'actions, produisent des trajectoires de différentes amplitudes. Comme illustré sur la Figure 4.2b, deux instances d'une même classe d'action produiront nécessairement la même forme globale mais pourront différer sensiblement en termes d'amplitudes des trajectoires associées.

Ceci est principalement dû aux différences de style de performance inhérentes à chaque sujet, conduisant à différentes amplitudes d'une même classe d'action. Dans certaines applications, la capture de telles *variabilités intra-classe* pourrait être souhaitable. En effet, elles peuvent apporter des informations supplémentaires et permettre différentes interprétations de la même classe d'action (pensez au réglage du volume audio en fonction de l'amplitude d'une action). Cependant, dans notre étude, ces variabilités sont indésirables et doivent donc être neutralisées.

4.2.2 Approche de détection d'actions 3D basée sur le déplacement curviligne : CuDi3D

L'approche OAD que nous proposons est illustrée dans la Figure 4.3. En entrée, un flot continu de données squelettiques corps-complet est reçu et une décision (pouvant être une des classes d'actions ou rien) est émise pour chaque frame d'entrée. L'approche repose sur trois étapes de manière à adresser les trois difficultés identifiées pour la OAD, à savoir

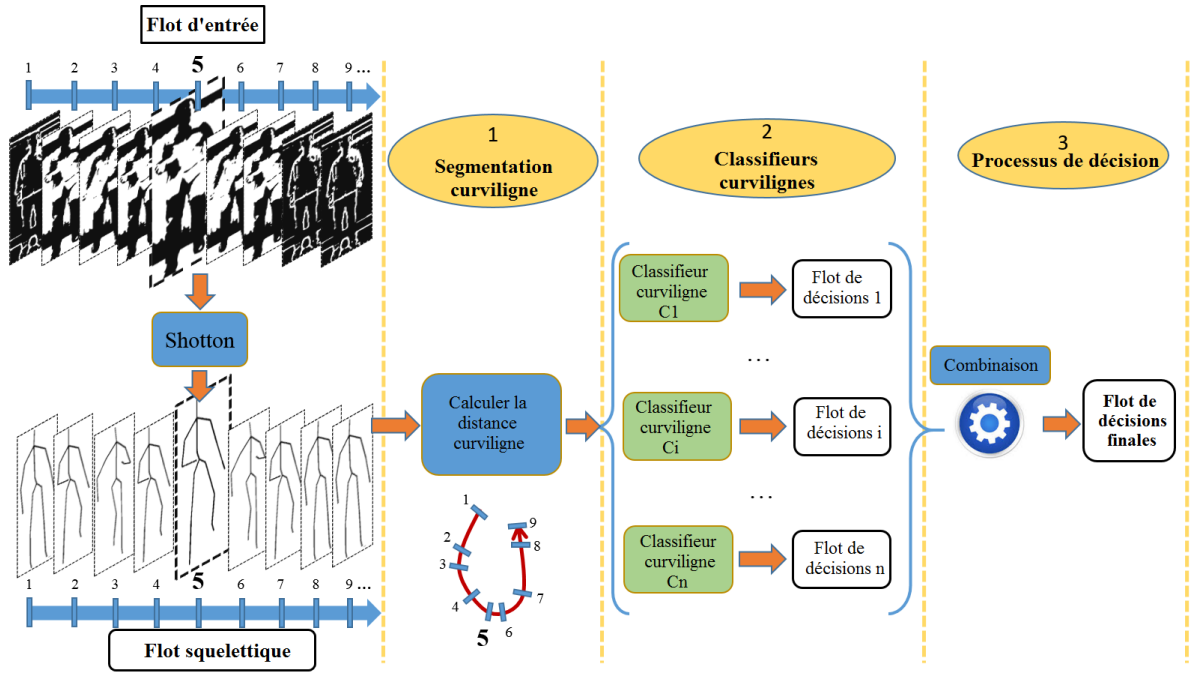

FIGURE 4.3 – Vue d'ensemble de l'approche OAD proposée. Cette approche est composée de trois étapes, de sorte qu'à chaque étape une des difficultés OAD est abordée. La *variabilité temporelle* est considérée à la première étape. La *variabilité spatiale inter-classes* est abordée à la deuxième étape. La *variabilité spatiale intra-classe* est abordée à la dernière étape.

variabilité temporelle, *variabilité spatiale inter-classes* et *variabilité spatiale intra-classe*.

Ci-après, nous détaillons les solutions pour adresser chacune des trois variabilités. Nous commençons d'abord par introduire le concept de la segmentation curviligne. Nous détaillons ensuite l'étape d'apprentissage des classifieurs basés sur une fenêtre curviligne. Enfin, nous expliquons le processus de décision qui permet de générer le flot de décision final.

4.2.2.1 Segmentation curviligne

Comme pour les approches de reconnaissance d'actions pré-segmentées, nous appliquons aux données squelettiques le prétraitement permettant de s'affranchir de la variabilité morphologique (cf. chapitre 3). Ainsi, pour chaque frame nous calculons les V vecteurs amorphologiques relatifs aux types d'actions à modéliser. A titre d'exemple, pour une action corps-complet, nous calculons $V = 4$ vecteurs amorphologiques correspondant aux deux bras et aux deux jambes. Les trajectoires résultant du déplacement de ces vecteurs au cours de la performance d'une action sont alors considérées pour construire la

représentation de cette action.

Pour faire face à la *variabilité temporelle*, nous proposons un nouveau concept basé sur une segmentation curviligne. Ce concept consiste à définir dynamiquement des fenêtres en fonction de la quantité d'informations (c'est-à-dire de mouvements) disponible dans le flot d'entrée. La taille de la fenêtre est ainsi contrainte par le déplacement curviligne du squelette au lieu d'être indexée au flot temporel habituel. L'adaptation de la taille des fenêtres curvilignes permet de capturer une même quantité d'informations à chaque frame. Ainsi, les représentations extraites sur les fenêtres curvilignes pendant les deux phases d'entraînement et de test sont cohérentes dans la mesure où elles représentent des (segments de) mouvements avec une quantité similaire d'informations quelle que soit la vitesse d'exécution.

Pour mieux comprendre ce nouveau concept de fenêtre curviligne, nous illustrons dans la Figure 4.4 la différence entre une fenêtre curviligne et une fenêtre glissante temporelle

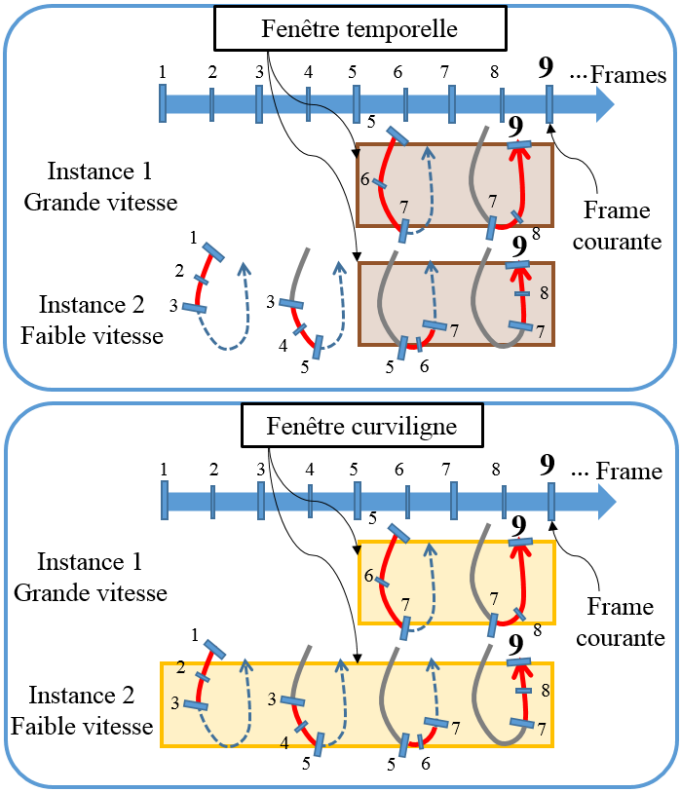

FIGURE 4.4 – Illustration de la différence entre une fenêtre curviligne et une fenêtre temporelle conventionnelle. Nous considérons le mouvement extrait avec ces deux fenêtres à la frame 9 pour deux instances de la même classe. Ces deux instances sont effectuées à des vitesses différentes. Le rectangle gris représente une fenêtre temporelle et le rectangle jaune représente la fenêtre curviligne.

classique sur un flot d'entrée squelettique. En particulier, nous montrons comment ces deux types de fenêtres fonctionnent sur deux instances différentes. Ces instances appartiennent à la même classe d'action mais sont exécutées à des vitesses différentes, d'où l'étalement sur des intervalles de temps différents.

Il est possible de noter que la fenêtre curviligne couvre des segments de mouvement produisant le même déplacement curviligne, indépendamment de leur vitesse d'exécution. En fait, seul le déplacement curviligne contrôle la taille de la fenêtre et non le nombre de frames nécessaires pour l'effectuer. Au contraire, la fenêtre glissante temporelle considère toujours le même nombre de frames quel que soit le déplacement curviligne résultant. Dans cette illustration, on peut remarquer que la fenêtre glissante temporelle ne couvre pas toujours une partie appropriée du mouvement car la taille de cette fenêtre est basée sur la temporalité. La fenêtre curviligne permet donc l'extraction de descripteurs sur des segments de mouvement cohérents, ce qui permettrait de mieux adresser la *variabilité temporelle*.

Formellement, nous utilisons le déplacement curviligne des articulations comme métrique de la quantité d'informations. Nous introduisons donc la fonction $CuDi(F_S, F_E)$ qui permet le calcul du déplacement curviligne pour un segment de mouvement donné, en partant de la frame F_S et en s'achevant à la frame F_E, comme suit :

$$CuDi(F_S, F_E) = \sum_{i=F_S}^{F_E} d_i^{Moy} \tag{4.1}$$

où d_i^{Moy} est le déplacement moyen instantané calculé pour chaque frame i et pour V trajectoires comme indiqué dans l'équation 4.2 :

$$d_i^{Moy} = \sqrt{\sum_{v=1}^{V} (d_i^v)^2} \tag{4.2}$$

où d_i^v est la valeur du déplacement du $v^{ème}$ vecteur squelettique pour $1 \leq v \leq V$, calculé comme suit :

$$d_i^v = \sqrt{(x_i^v - x_{i-1}^v)^2 + (y_i^v - y_{i-1}^v)^2 + (z_i^v - z_{i-1}^v)^2} \tag{4.3}$$

avec x_i^v, y_i^v, z_i^v, x_{i-1}^v,y_{i-1}^v, z_{i-1}^v les coordonnées 3D des $v^{ème}$ vecteurs correspondants dans la frame courante i et la frame précédente $i-1$, respectivement. Par exemple, pour une action corps-complet, ce déplacement est calculé pour quatre vecteurs correspondant aux deux bras et aux deux jambes.

Sur la base de la métrique définie dans l'équation 4.1, nous définissons une fenêtre curviligne comme une fenêtre glissante dont la taille temporelle est continuellement mise

à jour de sorte qu'elle englobe, à chaque frame, un mouvement ayant un déplacement curviligne spécifique. Par exemple, à une frame donnée F_t, la fenêtre curviligne devrait englober le segment de mouvement se terminant à cette frame F_t et commençant à la frame F_S. La frame F_S est déterminée de telle sorte que :

$$S = \max_{s_i} CuDi(F_{s_i}, F_t) \geq \rho \quad ; 1 \leq s_i < t \tag{4.4}$$

avec ρ un seuil de déplacement curviligne spécifique à chaque classifieur.

Cette fenêtre curviligne est utilisée pour traiter les données d'entrée et pour former des classifieurs curvilignes, comme expliqué dans la section suivante.

4.2.2.2 Classifieurs curvilignes

Le deuxième problème considéré dans notre étude est la *variabilité spatiale inter-classes*. Cela traduit le fait que différentes classes d'actions produiraient des déplacements curvilignes différents. Ainsi, une fenêtre curviligne permet d'adresser la *variabilité temporelle* d'une seule classe d'action, mais pas la *variabilité spatiale inter-classes*. Pour y pallier, nous proposons donc d'employer plusieurs tailles de fenêtres curvilignes et par conséquent autant de classifieurs qu'il y a de valeurs distinctes de déplacements curvilignes.

Nous notons les différents classifieurs Ci, et les différentes classes d'actions par Gi avec $1 \leq i \leq n$ et n le nombre total de classes d'actions (n est aussi le nombre total de classifieurs). Dans le cas général, il y aura autant de classifieurs que de classes d'actions car nous supposons que toutes les actions ont des déplacements curvilignes différents. En pratique, le nombre de classifieurs pourrait être réduit si certaines actions avaient des déplacements curvilignes similaires ce qui permettrait de réduire le temps de traitement. Par ailleurs, chacun de ces classifieurs est entraîné pour reconnaître toutes les classes d'actions [1] et pas uniquement pour reconnaître la classe à laquelle il est associé. Dans ce qui suit, nous considérons le cas général dans lequel il y a autant de classifieurs que de classes d'actions.

Pour former un classifieur donné Ci, nous calculons d'abord la quantité de déplacement curviligne $CuDi_i$ associée à la classe d'action Gi. Il s'agit de la moyenne de tous les déplacements curvilignes des instances appartenant à la classe Gi (cf. équation 4.5).

$$CuDi_i = \frac{1}{K} \times \sum_{k=1}^{K} CuDi(F_{S_k}, F_{E_k}) \tag{4.5}$$

1. A la condition que ces actions produisent au moins autant de déplacement curviligne que celui requis par le classifieur considéré.

où K est le nombre total d'instances appartenant à la classe d'action Gi. F_{S_k} et F_{E_k} sont respectivement le début et la fin de la $k^{ème}$ instance.

Le déplacement curviligne $CuDi(F_{S_k}, F_{E_k})$ de chaque instance k de l'ensemble d'apprentissage est calculé entre le début F_{S_k}et la fin F_{E_k} au moyen de l'équation 4.1. La fin de l'action correspond en fait au concept de *point d'action* introduit par [NS12]. En réalité, l'objectif étant de détecter les actions en temps réel, le point d'action ne correspond pas à la fin effective d'une action mais à l'instant auquel la présence de l'action est claire et peut être distinguée des autres classes et identifiée de façon unique.

Ensuite, au moyen de chaque fenêtre curviligne de taille $CuDi_i$, les instances de l'ensemble d'apprentissage de différentes classes d'actions sont analysées en partant de leur début jusqu'au point d'action. Au cours de ce processus, des descripteurs locaux sont extraits selon cette fenêtre curviligne pour construire l'ensemble d'apprentissage du classifieur Ci. Nous avons retenu comme descripteurs locaux les descripteurs **HIF3D** que nous avons conçus pour modéliser des trajectoires 3D et que nous avons introduits dans le chapitre 3. De plus, chacun des classifieurs est un SVM multi-classes entraîné suivant la stratégie un-contre-tous. Nous utilisons pour cela la bibliothèque LIBSVM [CL11b]. Cet apprentissage correspond à la deuxième étape du processus illustré dans la Figure 4.3.

4.2.2.3 Processus de décision

Le processus de décision fait référence à la manière dont les décisions de tous les classifieurs sont combinées lors du test pour détecter la classe finale. Au cours de ce processus, la troisième difficulté, à savoir la *variabilité spatiale intra-classe*, est considérée. Cette variabilité fait référence au fait que des instances appartenant à la même classe d'action peuvent entraîner des déplacements différents.

Ce processus est illustré dans la Figure 4.5, où on y voit un flot d'entrée traité avec plusieurs fenêtres curvilignes de tailles curvilignes différentes. Un classifieur donné Ci n'est sollicité que si le déplacement curviligne cumulé à la frame courante dépasse la taille curviligne $CuDi_i$ de la fenêtre curviligne associée à ce classifieur Ci. Les décisions locales $Predicted\_i$ de chaque classifieur Ci sont ensuite traitées et combinées au sein d'un système de fusion pour obtenir la classe finale.

En particulier, le système de fusion est composé de deux types de blocs de décision à savoir les blocs **B** et **C** (Figure 4.5). Chaque classifieur Ci est associé à un bloc **B** qui est utilisé pour traiter le flot de prédiction brut $Predicted\_i$ du classifieur. Chaque bloc **B** est basé sur un histogramme local qui a autant d'entrées qu'il y a de classes à prévoir. Cet histogramme local cumule le score de chaque classe prédite par le classifieur Ci. La sortie de chaque bloc **B** est un flot noté $Output\_i$. Ces flots de sortie sont ensuite fusionnés dans le bloc **C** pour obtenir le flot de sortie final (Figure 4.5). Le bloc **C** est basé sur

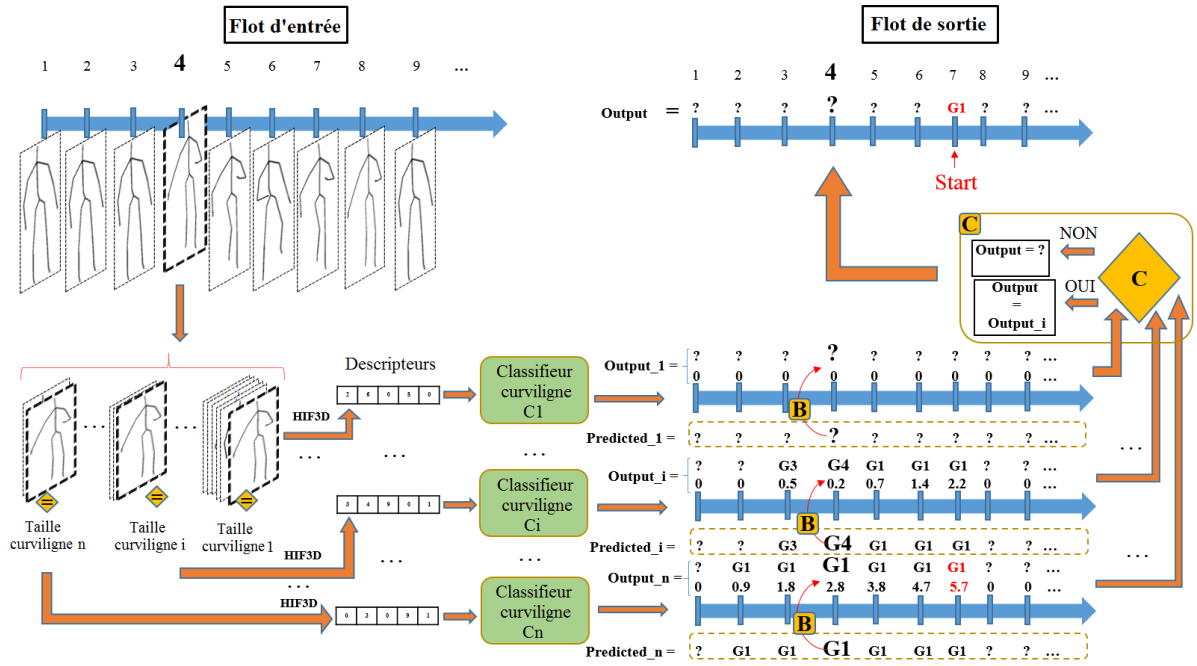

FIGURE 4.5 – Aperçu du fonctionnement global du système proposé. Ce système est composé de classifieurs curvilignes, un pour chaque taille curviligne. Différents classifieurs curvilignes extraient des descripteurs sur différentes fenêtres curvilignes. En outre, les blocs '**B**' et '**C**' ont en charge de traiter les classes prédites brutes et de combiner les décisions des différents classifieurs, respectivement. Le symbole '**?**' signifie qu'aucune classe n'est prédite, alors que G1, ..., Gn sont des classes et C1, ..., Cn sont des classifieurs. *Predicted_i* et *Output_i* sont respectivement le flot brut et le flot traité. Le flot *Output_i* est le flot *Predicted_i* plus les scores cumulés à chaque frame.

un histogramme global qui est continuellement mis à jour par les sorties de chaque bloc **B**. Cet histogramme global a autant d'entrées qu'il y a de classifieurs. Dans ce qui suit, nous expliquons d'abord comment un bloc **B** traite le flot de prédiction brut fourni par un classifieur *Ci*. Ensuite, nous considérons le fonctionnement du bloc **C** qui fusionne tous les flots de décisions locales.

Fonctionnement de l'histogramme local de chaque classifieur

Pour expliquer le fonctionnement de l'histogramme local, nous commençons d'abord par détailler l'aspect théorique avant de donner un exemple avec trois classifieurs dans la Figure 4.6.

Comme illustré dans la Figure 4.5 (partie gauche), chaque classifieur *Ci* reçoit un vecteur de descripteurs construit sur une fenêtre curviligne différente et se terminant à la frame courante. Le résultat brut de chaque classifieur, noté *predicted_i*, peut avoir

pour valeur le symbole ? ou l'une des classes d'actions ($G1$, $G2$, ...). Le symbole ? signifie qu'aucune décision n'a encore été prise, car le déplacement curviligne cumulé n'atteint pas encore la taille curviligne du classifieur en question. Le résultat $predicted\_i$ est ensuite passé au bloc **B** du classifieur Ci pour mettre à jour son histogramme local noté His_i. Cet histogramme a autant d'entrées qu'il y a de classes d'actions et est utilisé pour enregistrer le score de confiance cumulé de chaque classe potentielle.

De manière plus formelle, la $j^{ème}$ entrée ($1 \leq j \leq n$) d'un histogramme His_i, associé au classifieur Ci, est mise à jour à chaque instant selon l'expression suivante :

$$His_i(j) = \begin{cases} His_i(j) + \beta, & \text{si } j = Predicted\_i \\ His_i(j) - \gamma, & \text{sinon} \end{cases} \tag{4.6}$$

où β et γ ne sont pas des paramètres mais des différences de scores obtenus par chaque classifieur.

En fait, à chaque prédiction, le classifieur donne le score de confiance correspondant à la classe prédite mais également celui de toutes les autres classes. Les scores de confiance sont compris entre 0 à 1, de sorte que plus le classifieur est proche de 1, plus le classifieur est confiant de sa décision. β est égal à la différence entre le score de la classe actuellement prédite, $Predicted\_i$, et le score de la deuxième meilleure classe prédite par le classifieur Ci. γ correspond à la différence entre le score de $Predicted\_i$ et le score de la $j^{ème}$ entrée de l'histogramme. n est le nombre total de classes.

L'objectif principal de cette procédure est d'être plus robuste aux confusions entre deux classes différentes en retardant la décision finale. En fait, si les deux meilleures classes prédites par un classifieur ont des scores très proches, et donc la différence de score est faible, il est probable alors que le classifieur confonde ces deux classes car il n'a pas encore assez d'informations pour décider de l'action en cours. Par conséquent, l'histogramme local associé devra croître plus lentement en ajoutant la différence entre ces deux classes au lieu d'ajouter par exemple le score de confiance de la classe la plus probable. Ceci permet donc d'intégrer implicitement ce que l'on appelle communément **rejet de confusion** dans de telles procédures de décision.

Nous illustrons la procédure de mise à jour d'un histogramme local dans la Figure 4.6. Dans cette illustration, nous supposons qu'il existe trois classes d'actions, à savoir $G1$, $G2$ et $G3$ et nous considérons le fonctionnement de l'histogramme local pendant quatre frames (4, 5, 6 et 7). Le classifieur respectivement prédit $G1$, $G2$, $G1$ et $G1$ aux frames 4, 5, 6 et 7. Pour mettre à jour l'histogramme local d'un classifieur, le score cumulé de la classe prédite est augmenté par la différence de scores β tandis que celui des autres classes est diminué par les différences γ (cf. équation 4.6). Par exemple, à la frame 5 de la Figure 4.6, le classifieur a prédit la classe $G2$. Par conséquent, le score cumulé de $G2$ (barre

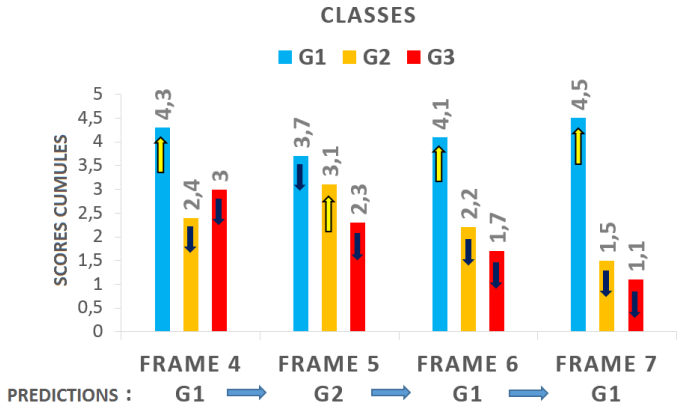

FIGURE 4.6 – Illustration du fonctionnement de l'histogramme local avec trois classes aux frames 4, 5, 6 et 7. ↑ et ↓ symbolisent respectivement une augmentation et une diminution du score.

du milieu) est augmenté. Simultanément, les scores des autres classes sont diminués. Au contraire, comme la classe prédite à la frame 6 est $G1$, le score cumulé de $G2$ est diminué ainsi que celui de $G3$ alors que le score de G1 est augmenté.

Ainsi, à chaque instant, la sortie du bloc **B** associé au classifieur Ci est la classe venant d'être prédite $Predicted\_i$ avec le score correspondant dans l'histogramme local His_i, à savoir $His_i(j = Predicted\_i)$. Ce flot est ensuite transmis au bloc **C** qui fusionne les décisions locales pour déterminer la classe finale. Le fonctionnement du bloc **C** est présenté ci-après.

Fonctionnement de l'histogramme global

À cette étape, le but est de combiner le flot de sortie de chaque classifieur pour obtenir la classe finale. Un histogramme global est alors utilisé, noté His_Global, qui est composé d'autant d'entrées qu'il y a de classifieurs. Chaque entrée i correspond à la classe $Predicted\_i$ prédite par le classifieur Ci à la frame courante avec son score cumulé, c'est-à-dire $His_i(j = Predicted\_i)$. L'idée de base de la fusion de décision est d'assurer un compromis entre, d'une part, le degré de confiance accordé à la classe détectée et, d'autre part, la latence de détection. A cette fin, nous calculons expérimentalement une matrice de seuils, notée *ThreshMat*, qui est utilisée au sein du bloc **C**. *ThreshMat* est une matrice $m \times n$ où m est le nombre de classifieurs et n le nombre de classes. Nous considérons ici le cas général où il y a autant de classifieurs que de classes, c'est-à-dire $m = n$.

De plus, la manière dont les valeurs de *ThreshMat* sont déterminées est cruciale pour assurer l'équilibre entre la réduction de la latence de détection et l'augmentation de la robustesse aux faux positifs. A cette fin, nous calculons d'abord deux matrices : *Precocity-*

Mat et *ConfusionMat*. *PrecocityMat* est composée de seuils cumulés qu'un classifieur donné pourrait idéalement atteindre pour chaque classe. Au contraire, *ConfusionMat* contient les seuils minimaux qu'un classifieur doit dépasser pour éviter de confondre des classes différentes.

En pratique, pour la calibration des deux matrices, nous ne considérons que l'ensemble d'apprentissage. Nous appliquons à cet ensemble d'apprentissage une procédure de validation croisée afin d'obtenir un nouvel ensemble d'apprentissage et un ensemble de validation. En particulier, nous divisons l'ensemble d'apprentissage principal en 10 groupes, ensuite nous entraînons chaque classifieur sur 9 groupes et puis nous validons sur le groupe restant. Nous répétons cette opération 10 fois et à chaque itération, nous simulons le processus de détection entre le début et le point d'action de chaque échantillon du groupe de validation. Nous reportons ensuite dans le $(i,j)^{ème}$ élément de la matrice *PrecocityMat* la moyenne des scores cumulés jusqu'au point d'action par le classifieur *Ci* pour la classe *Gj* si la classe prédite *Gj* correspond au label réel de la séquence traitée. En même temps, nous mettons à jour la matrice *ConfusionMat* de sorte que son $(i,j)^{ème}$ élément est la moyenne des scores cumulés par le classifieur *Ci* pour la classe *Gj* si cette classe prédite (*Gj*) est fausse (c'est-à-dire différente du label réel de la séquence).

Enfin, chaque élément $\theta_{i,j}$ de la matrice *ThreshMat* est obtenu comme suit :

$$\theta_{i,j} = \frac{PrecocityMat(i,j) + ConfusionMat(i,j)}{2} \tag{4.7}$$

D'une part, grâce à la contribution de la matrice $ConfusionMat$ cette procédure évite les décisions trop rapides qui pourraient engendrer la confusion avec d'autres classes. Ceci peut se produire quand les frames précédemment observées ne sont pas suffisantes pour permettre aux classifieurs de décider de l'action en cours. D'autre part, grâce à la contribution de la matrice $PrecocityMat$ cette procédure vise à réduire la latence en émettant des décisions dès qu'un classifieur devient suffisamment confiant quant à sa décision. Il est à noter que cette procédure de calibration permet de calculer automatiquement des seuils à partir d'une base d'apprentissage indépendamment des actions à reconnaître et peut donc être appliquée à tous les types de bases de données.

Lors de la fusion des décisions, une classe *Gj* prédite par un classifieur *Ci* est considérée comme la classe finale si le score cumulé par ce classifieur *Ci* pour cette classe *Gj* dépasse le seuil $\theta_{i,j} = ThreshMat(i,j)$. Ainsi, tout classifieur entraîné avec une taille de fenêtre curviligne spécifique est capable de prédire n'importe quelle classe si ce classifieur est suffisamment sûr de sa décision. En procédant de cette manière, nous fournissons une solution pour adresser la *variabilité spatiale intra-classe*, étant donné qu'un classifieur peut détecter l'occurrence d'une action autre que sa classe d'action de base, c'est-à-dire celle correspondant à sa taille curviligne. Nous résumons le processus de décision global

dans l'équation 4.8.

$$Output = \begin{cases} \mathbf{Gj} \quad , \quad \text{si} \quad \exists \quad 1 \leq i,j \leq n \quad \& \\ \qquad\qquad\quad His\_Global(i) \geq \theta_{i,j} \quad \& \\ \qquad\qquad\quad Output\_i = Gj \\ ? \quad , \quad \text{sinon} \end{cases} \tag{4.8}$$

Ainsi, la décision finale est égale à un ? tant que les décisions locales ne sont pas prises, ce qui correspond au fait qu'aucune action n'est détectée. De plus, si le système de fusion émet deux classes potentielles ou plus, la décision est reportée jusqu'à ce qu'il ne reste qu'une seule classe. Dès qu'une classe est retenue, le début du cumul de son score ainsi que la taille curviligne du classifieur ayant permis sa détection sont utilisés pour déterminer le début de cette action, c'est-à-dire la segmentation du flot. Nous considérons en effet que le début d'action correspond à l'instant du début du cumul du score de la classe auquel on soustrait temporellement la taille de la fenêtre curviligne du classifieur. L'instant de fin de l'action (ou son point d'action) correspond à l'instant final de détection de cette action par le classifieur. De plus, à la fin du processus de décision, tous les histogrammes et les déplacements curvilignes cumulés pour chaque classifieur sont réinitialisés à zéro. Cette réinitialisation permet en fait de préparer le système de décision pour la détection d'une prochaine action potentielle dans le flot d'entrée. C'est de cette manière que nous gérons d'une part la détection des actions et d'autre part la segmentation du flot d'entrée.

Une illustration du fonctionnement de l'histogramme global du bloc **C** est proposée

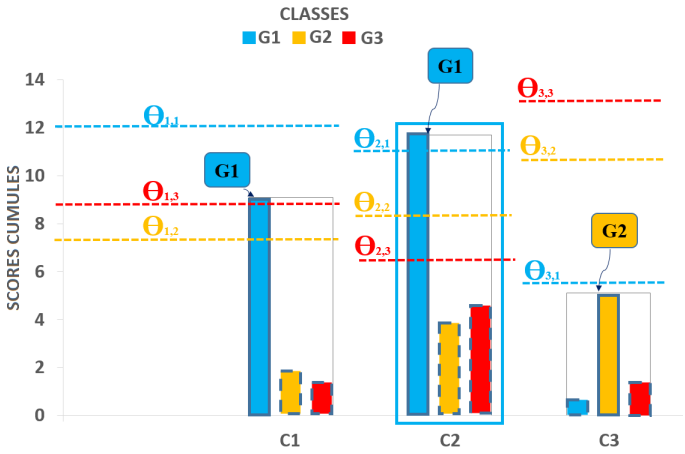

FIGURE 4.7 – Illustration du fonctionnement de l'histogramme global à la frame 7 avec trois classifieurs C1, C2 et C3 qui peuvent prédire la classe G1, G2 ou G3. Chaque classifieur Ci possède trois seuils $\theta_{i,1}$, $\theta_{i,2}$ et $\theta_{i,3}$ relatifs à la prédiction des trois classes G1, G2 et G3.

dans la Figure 4.7. En suivant l'illustration de la Figure 4.6, nous considérons ici trois classifieurs C1, C2 et C3 correspondant respectivement à la taille curviligne de trois classes : G1, G2 et G3. Dans la Figure 4.7, nous présentons l'état de l'histogramme global à la frame 7. La classe prédite par les classifieurs C1 et C2 est G1, avec des scores cumulés différents. La classe prédite par le classifieur C3 est G2. Le classifieur $C2$ est celui qui a réussi à détecter l'occurrence de la classe d'action $G1$. En fait, le score cumulé par le classifieur $C2$ pour la classe $G1$ est le seul score qui dépasse le seuil associé, c'est-à-dire $\theta_{2,1}$. Ainsi, la classe prédite finale est $G1$.

4.3 Extension de l'approche CuDi3D à des problématiques connexes

Dans cette section, l'idée est de se baser sur l'approche **CuDi3D**, en particulier le concept de fenêtre curviligne, pour d'une part **reconsidérer** le problème de la **reconnaissance d'actions 3D pré-segmentées** que nous avons abordé dans le chapitre 3 et d'autre part **considérer** le dernier problème, plus complexe, de détection précoce.

4.3.1 Reconnaissance d'actions 3D pré-segmentées

L'approche **CuDi3D** que nous avons conçue à des fins de détection d'actions dans un flot non segmenté peut être adaptée pour résoudre plus efficacement le problème de reconnaissance d'actions pré-segmentées que nous avons abordé dans le chapitre 3. Nous rappelons d'abord que ce problème consiste à étiqueter une seule action pré-segmentée après que cette action soit complètement achevée. Néanmoins, même dans ce cas de figure, la segmentation est souvent faite manuellement et donc le début et la fin ne sont pas précisément connus, d'où l'intérêt d'adapter la **CuDi3D** à la reconnaissance d'actions pré-segmentées.

Une différence essentielle par rapport au cas de reconnaissance d'actions pré-segmentées que nous avons abordé dans le chapitre 3 est que la séquence d'action est traitée progressivement, c'est-à-dire, frame par frame, comme pour le cas en-ligne. Par rapport à la détection dans un flot non segmenté, la tâche de reconnaissance pré-segmentée est allégée étant donné que les instants de début et de fin sont connus à l'avance. Par conséquent, certains changements sont apportés à l'approche **CuDi3D** afin de profiter de cette nouvelle information.

En ce qui concerne l'étape d'apprentissage, les classifieurs curvilignes tels que présentés dans les sections 4.2.2.1 et 4.2.2.2 sont entraînés de la même manière. En outre, nous ajoutons un nouveau classifieur qui est entraîné avec l'ensemble des actions pré-segmentées

de toutes les classes. Ce classifieur supplémentaire ne sera utilisé que si aucun classifieur curviligne n'est lancé comme expliqué ci-dessous. Au cours de l'étape de test, le processus de décision est mis à jour et est illustré dans la Figure 4.8.

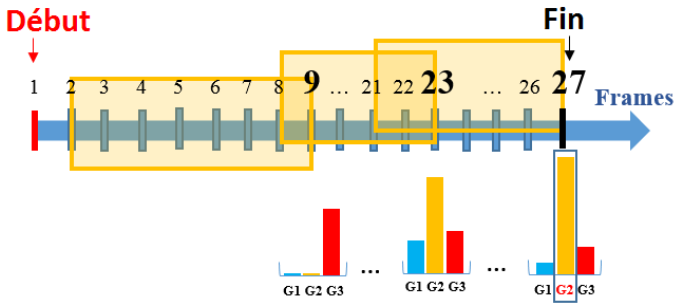

FIGURE 4.8 – Processus de décision adapté pour la reconnaissance d'actions pré-segmentées.

En particulier, à chaque frame de la séquence de test, un seul classifieur est lancé, au lieu de plusieurs classifieurs en parallèle, puisque nous connaissons exactement le début de l'action en cours. A partir du début de la séquence de test, le déplacement curviligne cumulé est calculé à chaque frame F_t et si ce déplacement dépasse la taille curviligne $CuDi_i$ d'un classifieur donné C_i alors ce classifieur est activé. Parmi les classifieurs activés C_i à la frame courante F_t, nous considérons seulement le classifieur C_j qui correspond à la plus grande fenêtre curviligne $CuDi_j$ (cf. équation 4.9).

$$CuDi_j = \max_{F_t} CuDi_i \tag{4.9}$$

Les descripteurs **HIF3D** ne sont ensuite extraits que selon la fenêtre curviligne correspondante $CuDi_j$. Le but d'utiliser à chaque frame un seul classifieur activé, celui avec la plus grande taille curviligne, est double. D'une part, cela nous permet de considérer à chaque frame le plus grand segment de mouvement afin de prendre en compte autant d'informations que possible. D'autre part, cela nous permet de ne considérer qu'un segment de mouvement approprié (et pas nécessairement toute la séquence). Ceci garantit notamment une cohérence entre ce segment de mouvement et ceux utilisés pour l'apprentissage des classifieurs curvilignes.

De plus, au fur et à mesure que la séquence de test est traitée, un histogramme est mis à jour à chaque frame avec la décision de sortie et le score de confiance émis par le classifieur (Figure 4.8). En fait, pour la reconnaissance d'actions pré-segmentées, un seul histogramme est utilisé, similaire aux histogrammes locaux présentés dans la section 4.2.2.3. Néanmoins, cet histogramme est global car il est mis à jour avec la sortie de tous les classifieurs. A chaque fois qu'une classe Ci est détectée par le classifieur utilisé

Cj, la valeur de l'histogramme correspondant à la $i^{ème}$ entrée est incrémentée avec la différence entre le score de la classe prédite Ci et le score de la seconde classe prédite par le classifieur Cj. Simultanément, les scores de toutes les autres classes sont diminués selon l'équation 4.6. A la fin de la séquence traitée, la classe finale est celle associée au score le plus élevé de l'histogramme. Comme il n'y a qu'une seule action, il n'y pas de risque de confusion entre plusieurs actions partageant des débuts communs. De ce fait, les matrices des seuils de confiance ne sont plus apprises. Si aucun classifieur n'a été lancé, en raison d'un déplacement curviligne insuffisant de la séquence, alors les descripteurs sont extraits sur toute la séquence et le classifieur supplémentaire, présenté ci-dessus, est utilisé pour obtenir la classe finale.

4.3.2 Détection précoce d'actions 3D

La détection précoce d'actions vise à déterminer, au plus tôt, la classe d'une action si elle a lieu, en utilisant le moins d'observations possibles. La plupart des travaux existants [Ryo11, CBB+13, DT06, LF12, LCS14], qui se sont intéressés à la problématique d'identification précoce d'une action, l'ont fait sur des séquences pré-segmentées (une seule action à reconnaître dans la séquence traitée). L'évaluation est alors effectuée au moyen du calcul du rapport entre les instants de début/fin prédits et ceux issus de la vérité terrain. Suivant les protocoles d'évaluation, ces rapports varient de 0.1 à 1 ; autrement dit, un rapport d'observation de 0.5 signifie une correspondance à partir de 50% de l'action, alors qu'un rapport de 1 correspond à la tâche de reconnaissance classique d'actions pré-segmentées avec une utilisation de toute l'action.

Notre objectif étant de tendre vers des applications d'interaction temps réel, nous nous intéressons surtout à la détection au plus tôt dans des séquences non segmentées. A cette fin, nous proposons une approche de détection précoce, dénommée **E-CuDi3D** pour *Early CuDi3D*, qui est basée sur l'approche **CuDi3D** présentée dans la section 4.2.2. L'idée principale de l'approche **E-CuDi3D** est d'adresser les différentes variabilités identifiées pour l'approche **CuDi3D** mais en en dérivant une approche au plus tôt.

En particulier, l'approche de détection précoce **E-CuDi3D** est constituée de trois modèles curvilignes. Ces modèles curvilignes permettent respectivement le traitement du flot d'entrée à court, moyen et long terme (cf. Figure 4.9). Chacun de ces modèles est en fait une variante de l'approche **CuDi3D**. Comme expliqué dans la section 4.2.2, chaque classifieur composant chaque modèle est associé à une classe d'action de manière à ce que la taille de la fenêtre qu'il utilise s'obtient en calculant la moyenne des déplacements curvilignes de toutes les instances de cette classe d'action. Ainsi, pour le premier modèle (modèle à long terme), les tailles curvilignes utilisées sont égales à 100% de ces valeurs

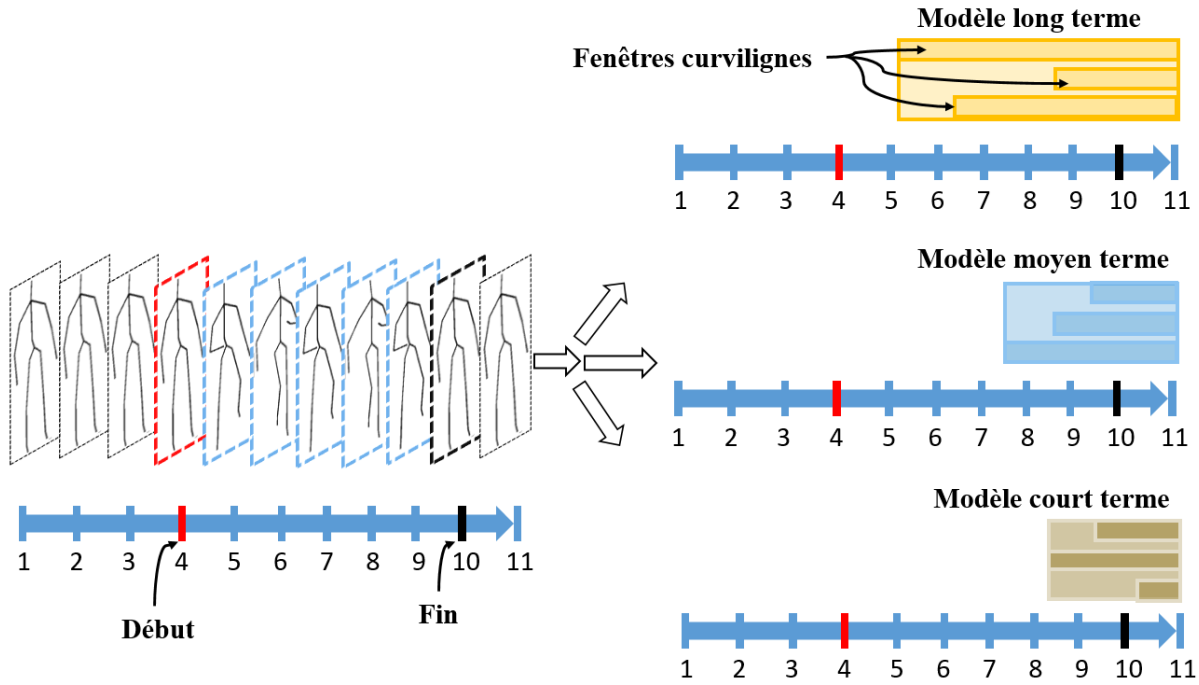

FIGURE 4.9 – Illustration du traitement d'un flot d'entrée par les trois modèles curvilignes à court, moyen et long terme.

moyennes (c'est l'approche **CuDi3D**). Pour le deuxième modèle (modèle à moyen terme), les tailles curvilignes utilisées sont égales à 50% des valeurs moyennes. Enfin, pour le dernier modèle (modèle à court terme), les tailles curvilignes utilisées sont égales à 10% des valeurs moyennes.

Considérons à présent la combinaison des décisions qu'émet chacun des trois modèles. La procédure de combinaison dépend en réalité de la finalité et du sens donné à la notion de détection précoce. En effet, nous nuançons dans nos travaux deux protocoles d'évaluation possibles de la détection précoce.

D'une part, le premier protocole consiste à procéder exactement comme pour les approches OAD avec l'objectif de le faire au plus tôt. D'autre part, suivant le deuxième protocole, l'évaluation est menée frame par frame indépendamment ; autrement dit les fausses détections qui surviennent souvent au début de l'action ne pénalisent pas l'évaluation de la performance sur les frames suivantes. Ce deuxième protocole est en fait une version très simplifiée de la détection précoce (plus simple même que la OAD) et c'est suivant ce protocole que certains travaux existants procèdent. En fait, le premier protocole est beaucoup plus contraignant (aussi bien par rapport au deuxième protocole que par rapport aux approches de la OAD) car, tout comme pour une approche de la OAD, dès qu'une mauvaise détection est annoncée, elle est comptabilisée dans le calcul de la

performance globale comme étant un faux positif. Néanmoins, ce protocole est plus proche d'une exploitation réaliste. Par exemple, pour une interface à base d'actions 3D où chaque détection enclenche une commande, il est impératif d'émettre au plus tôt **une seule** décision pour chaque performance mais pas pour chaque frame. Dans le cas contraire, les commandes vont se démultiplier ce qui nuirait à la finalité d'une telle interface de commande. Nous présentons donc ci-après la procédure de combinaison la plus appropriée pour chacun de ces deux protocoles.

Pour ce qui est du premier protocole, où l'objectif est **d'être le plus précoce possible en réduisant le risque de faux positifs**, la spécialisation de la procédure de combinaison se base sur la manière de déterminer les valeurs des seuils de la matrice *ThreshMat* pour chaque modèle. Comme expliqué plus haut dans la section 4.2.2, la matrice des seuils *ThreshMat* est une combinaison linéaire des matrices de confusion et de précocité précédemment définies. Cette combinaison est néanmoins différente pour chacun des modèles. En fait, pour assurer une équité entre ces trois modèles, la combinaison est inversement proportionnelle au pourcentage des tailles curvilignes défini pour chacun des modèles. Ceci correspond au fait que plus la taille curviligne d'un modèle est importante, moins grand sera le score qu'il devra cumuler pour se prononcer. Dans le cas contraire, seul le classifieur de faible pourcentage de distances curvilignes se prononcera. Nous proposons alors à ce que les valeurs $\theta_{i,j} = ThreshMat(i,j)$ soient déterminées comme suit :

$$\theta_{i,j} = \gamma_p \times PrecocityMat(i,j) + (1 - \gamma_p) \times ConfusionMat(i,j) \tag{4.10}$$

où $\gamma_p = 1/4,\ 1/2,\ 3/4$ pour respectivement $p = 1, 2, 3$. Comme mentionné plus haut, les modèles d'indice $p = 1, 2, 3$, correspondent respectivement aux approches utilisant $\alpha = 100, 50, 10\%$.

Ainsi, comme pour une combinaison de classifieurs avec un seul modèle, la décision finale n'est pas prise tant qu'aucun modèle n'en a émis une. De plus, si les modèles émettent des décisions différentes, nous attendons jusqu'à ce qu'il n'en reste qu'une seule. Cette procédure permet de mettre en place des options de rejet pour éviter les faux positifs. Une fois que la décision finale est prise, tous les scores sont remis à zéro ainsi que les distances curvilignes cumulées par les modèles afin de préparer le système à la détection d'une potentielle prochaine action.

S'agissant à présent du second protocole, celui de la **détection précoce simplifiée**, la contrainte relative à la robustesse aux faux positifs n'a plus lieu d'être. En particulier, il n'y a plus besoin d'attendre qu'un classifieur soit suffisamment sûr de la classe qu'il identifie avant d'émettre une décision finale. De ce fait, les valeurs $\theta_{i,j}$ des matrices $ThreshMat$ regroupant les seuils de confiances de chacun des classifieurs Ci pour chaque classe Gj sont mises à zero et ce pour les trois modèles introduits précédemment.

De plus, nous avons opté pour une combinaison simple des décisions fournies par les trois modèles constituant notre approche. Cette combinaison est résumée dans l'équation ci après :

$$Output = \begin{cases} \textbf{Gi} \quad , \quad \text{si} \quad \exists \quad 1 \leq j \neq k \leq 3 \quad \& \\ \qquad\qquad\quad Output\_j = Gi \quad \& \\ \qquad\qquad\quad Output\_k = Gi \\ \textbf{Gi} \quad , \quad \text{sinon si} \quad Output\_3 = Gi \\ ? \quad , \quad \text{sinon} \end{cases} \tag{4.11}$$

Le principe de combinaison résumé dans l'équation 4.11 traduit le fait que si au moins deux modèles prédisent la même classe d'action **Gi** alors la frame est labellisée avec cette classe. Au contraire, s'il n'y a pas de consensus entre au moins deux modèles, alors la décision finale prise est celle émise par le modèle de plus petite taille curviligne. Ce choix est motivé par une volonté de précocité. L'idée est que s'il n'y a pas de consensus c'est probablement parce que l'action est à son début et c'est donc au classifieur à 10% de décider, les autres classifieurs n'ayant pas assez d'informations pour affiner leur réponse. Bien que nous procédons dans un cas non segmenté, et contrairement au fonctionnement standard des modèles, les distances curvilignes cumulées ne sont pas remises à zero à chaque nouvelle détection. Ceci est dû à l'allégement des contraintes dans ce deuxième protocole où l'objectif n'est pas de segmenter le flot en entrée mais d'étiqueter chaque frame indépendamment des autres.

4.4 Résultats expérimentaux et discussion

Nous présentons d'abord les résultats obtenus par l'approche **CuDi3D** dans le cadre de la détection en-ligne d'actions dans un flot **non segmenté** sur trois bases de données. Nous présentons ensuite les résultats des extensions de cette approche dans le cadre de la reconnaissance d'actions **pré-segmentées** et celui de la détection précoce d'actions **non segmentées**.

4.4.1 Résultats de l'approche CuDi3D

Nous présentons les résultats de notre approche dans le contexte de la détection d'actions en-ligne (OAD) dans un flot d'actions non segmenté. En particulier, l'approche **CuDi3D** est évaluée suivant différents protocoles et comparée avec une douzaine d'approches OAD de l'état de l'art [MHT15, FMKN12, STHES15, ZLP+13, MHT16, BAM13,

BMA12, BMA14, LLX+16, HYWDLT14, EMS16, DBP+17] sur trois bases de données squelettiques, dont **MSRC-12** [FMKN12], **G3D** [BMA12] et **MAD** [HYWDLT14].

4.4.1.1 Base de données MSRC-12

La base de données Microsoft Research Cambridge-12 (MSRC-12) contient des séquences de données squelettiques, consistant en 20 positions articulaires [FMKN12]. Elle comprend 12 classes d'actions effectuées par 30 sujets pour un total de 594 séquences (environ 50 séquences par classe). Une seule action est effectuée plusieurs fois le long de chacune des séquences. Lors de la collecte de cette base de données, les participants ont reçu les consignes de performances des actions suivant cinq modalités : Images, Texte, Vidéo, Images + Texte et Vidéo + Texte. Cette base est annotée avec des points d'actions [2].

Les expériences sur cette base de données sont conduites en trois étapes : d'abord, nous comparons l'approche proposée aux méthodes OAD précédentes en utilisant le protocole leave-subjects-out comme suggéré dans de nombreux travaux précédents [FMKN12]. Deuxièmement, afin de mieux estimer l'apport des fenêtres curvilignes, nous proposons de substituer dans notre approche ces fenêtres curvilignes par des fenêtres temporelles classiques tout en gardant les autres composants inchangés. Enfin, nous fournissons les résultats d'autres variantes de notre approche en faisant notamment varier le nombre de fenêtres curvilignes ou leurs tailles curvilignes.

Nous évaluons l'approche **CuDi3D** dans le cadre de la détection en-ligne d'actions en utilisant le protocole leave-subjects-out. C'est le protocole OAD le plus utilisé sur la base de données MSRC-12 et il consiste à diviser cette base en 10 ensembles disjoints. Chaque ensemble comporte toutes les réalisations d'un sous-groupe de sujets. À chaque itération, neuf ensembles sont utilisés pour l'apprentissage et un ensemble est utilisé pour les tests. La performance du système est mesurée en termes de précision et de rappel. La précision indique la fréquence à laquelle une action est réellement présente lorsque le système le prétend. Le rappel mesure le nombre d'actions correctement reconnues par le système. Pour chaque classe d'action, nous combinons les deux mesures pour calculer le F_{score} défini comme suit :

$$F_{score} = 2 * \frac{Precision * Rappel}{Precision + Rappel} \tag{4.12}$$

Vu qu'il existe plusieurs classes d'actions, nous avons calculé une moyenne des F_{score} sur toutes ces classes. Nous obtenons finalement cinq F_{score}, un pour chaque modalité de

2. Les instants auxquels la présence de l'action est claire et peut être distinguée des autres classes et identifiée de façon unique

consignes. Chaque F_{score} est une moyenne sur les 10 répétitions et les 12 classes d'actions. Par ailleurs une détection est comptabilisée comme étant positive si elle est émise au maximum 10 frames après le point d'action. Ceci correspond à une latence de 0.333 secondes. Ce même protocole a été utilisé pour l'évaluation des cinq approches suivantes : [FMKN12, ZLP$^+$13, STHES15, MHT15, MHT16].

Grâce à [HTGES13], la base MSRC-12 a également été annotée avec les instants de début et de fin. Nous nous sommes donc basés sur ces annotations pour calculer les tailles des fenêtres curvilignes. Nous avons ensuite entraîné douze (12) classifieurs curvilignes (un avec la distance curviligne de chaque classe) en faisant glisser à chaque fois, sur toutes les données d'apprentissage, une fenêtre curviligne contrainte par la taille curviligne correspondante. Nos résultats, ainsi que ceux obtenus par les approches OAD squelettiques antérieures, sont reportés dans la Table 4.1.

Méthodes	ELS [MHT15]	RF [FMKN12]	RTMS [STHES15]	SSS [ZLP$^+$13]	ELS [MHT16]	**CuDi3D**
Video + Text	0.6450	0.6790	0.7130	0.7070	0.7900	**0.8540** $\pm$ 0.0670
Images + Text	0.5810	0.5630	0.6560	0.7300	0.7110	**0.7530** $\pm$ 0.0880
Text	0.4370	0.4790	0.5210	**0.7130**	0.6220	0.6730 $\pm$ 0.1020
Video	0.5800	0.6270	0.6350	0.5570	0.7260	**0.8450** $\pm$ 0.0790
Images	0.4970	0.5490	0.5960	0.6660	0.6700	**0.7310** $\pm$ 0.1220
Total	0.5480	0.5790	0.6240	0.6750	0.7040	**0.7710**

TABLE 4.1 – OAD, MSRC-12 : Résultats de l'approche **CuDi3D** et ceux obtenus par les approches de l'état de l'art sur la base de données MSRC-12. L'évaluation est menée dans le contexte OAD, suivant le protocole leave-subjects-out avec une latence de $\Delta = 333ms$. La moyenne F_{score} et son écart-type sont indiqués pour chaque modalité d'instruction. ELS = Efficient Linear Search ; RF = Random Forests ; RTMS = Real-Time Multi-Scale ; SSS = Structured Streaming Skeleton.

Comme montré dans la Table 4.1, la meilleure performance sur cette base de données a été précédemment obtenue par une approche OAD simple mais efficace proposée par Meshry et al. [MHT16]. Leur approche, dénommée ELS, analyse le flot d'entrée avec une fenêtre glissante temporelle classique et recherche à chaque nouvelle frame le sous-intervalle pour lequel le classifieur donne un score maximal. Dans notre approche, l'objectif consiste également à rechercher un segment de mouvement qui correspond le mieux à un segment pré-entraîné. Néanmoins, notre approche fonctionnant avec des fenêtres curvilignes permet de mieux gérer les *variabilités temporelles*. Cela permet de réduire les faux positifs car l'approche n'émet pas de décisions lorsqu'il y a trop peu de mouvement. En outre, l'approche opère toujours sur des segments de mouvement de tailles curvilignes

pré-définies ce qui les rend homogène à la quantité d'informations apprise indépendamment des vitesses d'exécution des sujets. Ceci permet notamment d'augmenter la capacité de détection et d'atteindre une performance de **77.10**%. Il est important de noter que l'écart entre les résultats de notre approche et ceux obtenus par les meilleures approches de l'état de l'art est très significatif suivant presque toutes les modalités de consignes, sauf avec la modalité *Text* (allant de +**2.30%** en utilisant des Images + Texte à +**12.00%** en utilisant la Vidéo). En fait, l'utilisation de différentes modalités augmente les *variabilités intra* et *inter-classes* et les résultats obtenus suggèrent que notre approche adresse mieux ces variabilités.

Pour avoir une idée plus détaillée de la latence de détection de l'approche **CuDi3D**, nous fournissons dans la Figure 4.10 une courbe qui rapporte le F_{score} cumulé en fonction des instants de détection mesurés comme distances par rapport aux points d'actions. Comme suggéré dans le protocole de test, seules les détections survenant au maximum 10 frames après le point d'action sont prises en compte, ce qui correspond à une latence de $\Delta = 333ms$. Nous rapportons cependant les résultats de détection se produisant au-delà de cette limite afin de montrer si la performance de détection globale est améliorée lorsque

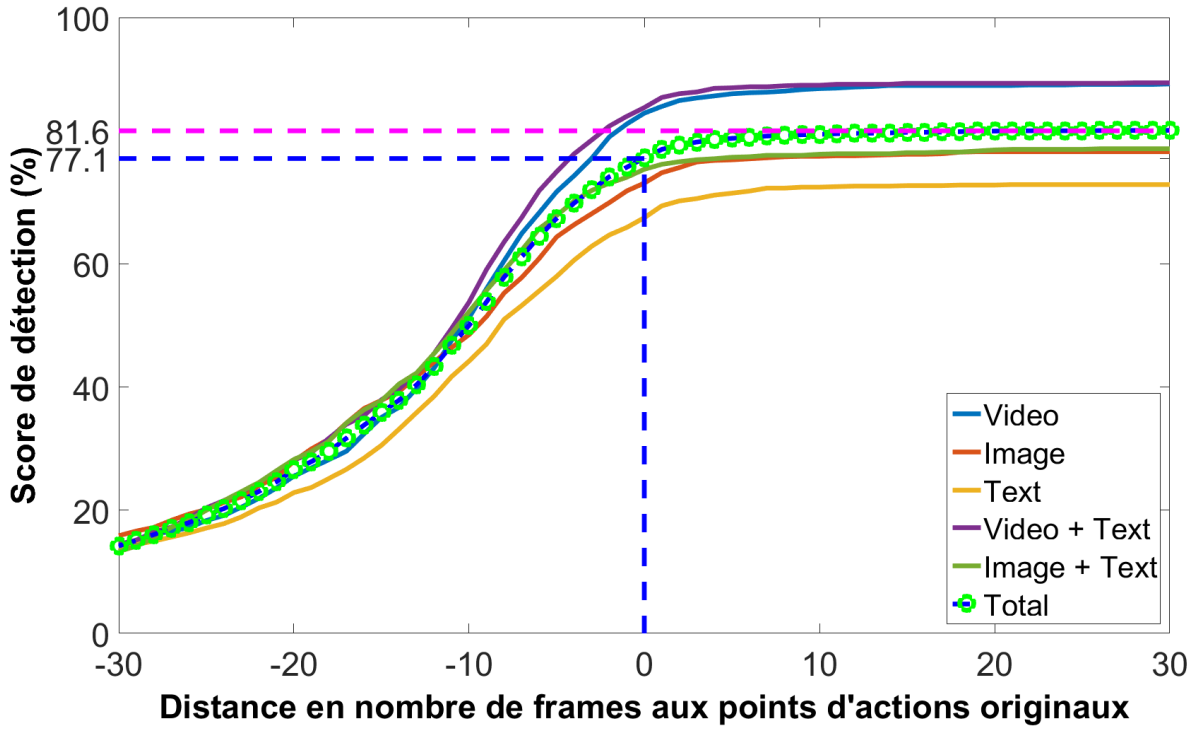

FIGURE 4.10 – OAD, MSRC-12 : Courbe cumulée des scores de détection en fonction de la distance au point d'action. 0 correspond au fait que le point d'action utilisé est celui initialement fourni avec la base.

la contrainte de latence est relâchée.

On peut d'abord remarquer que la plupart des détections se sont produites autour des instants des points d'actions, témoignant de la grande précision de notre approche et de sa capacité à détecter des actions avec des variations de vitesses d'exécution. De plus, nous pouvons voir que le score de détection obtenu selon le protocole leave-subjects-out, à savoir 77.10%, peut encore être amélioré si la contrainte concernant la limite supérieure de +10 frames après le point d'action est relâchée. En fait, le protocole proposé dans [FMKN12] considère qu'une action détectée au-delà du point d'action + 10 frames est un faux positif. Cependant, en raison justement des variabilités des vitesses d'exécution, certaines approches pourraient être pénalisées. Nous fournissons donc comme information complémentaire le score de détection maximum, à savoir **81.60%**, que notre approche est susceptible d'obtenir si une telle contrainte est assouplie.

En outre, pour illustrer quelques difficultés de détection, deux cas d'erreurs issus de la base MSRC-12 sont présentés dans la Figure 4.11. Pour illustrer d'abord la *variabilité inter-classes* qui provoque un échec de reconnaissance, nous avons sélectionné quelques frames d'une action de classe référencée comme G9 (première ligne) et une action d'une autre classe mais qui est fortement similaire, référencée comme G1 (ligne du milieu). Même si ces actions sont étiquetées différemment, on peut voir que ces deux mouvements sont presque similaires. De telles similitudes entre deux classes différentes rendent difficile, voire impossible, une détection correcte. Deuxièmement, des frames d'une autre instance de classe G1 sont reportées dans la rangée du bas. Ces frames sont réalisées par un sujet différent de celles présentées dans la rangée du milieu. Dans ce cas, on peut voir qu'il existe une différence substantielle entre les deux instances appartenant pourtant à la même classe. Cette *variabilité intra-classe* rend plus difficile la reconnaissance correcte de telles instances dans un cas non segmenté.

La deuxième expérience vise à mettre en évidence l'apport des fenêtres curvilignes par rapport aux fenêtres glissantes temporelles classiques. Nous avons donc reproduit la première expérience menée sur la base MSRC-12 en utilisant une variante de notre approche. Dans cette variante, les fenêtres curvilignes sont remplacées par des fenêtres glissantes temporelles, les autres composants restent inchangés. Nous avons utilisé autant de fenêtres glissantes temporelles que de classes et la même procédure est suivie pour déterminer leur taille temporelle ainsi que les seuils de confiance $\theta_{i,j}$. Les résultats sont rapportés dans la Table 4.2. Nous rapportons également dans cette table les résultats de l'approche **CuDi3D** pour faciliter la comparaison.

Sur la base des résultats présentés dans la Table 4.2, deux conclusions principales peuvent être tirées. D'une part, il apparaît clairement que l'utilisation de fenêtres curvilignes à la place des fenêtres temporelles permet une amélioration significative des per-

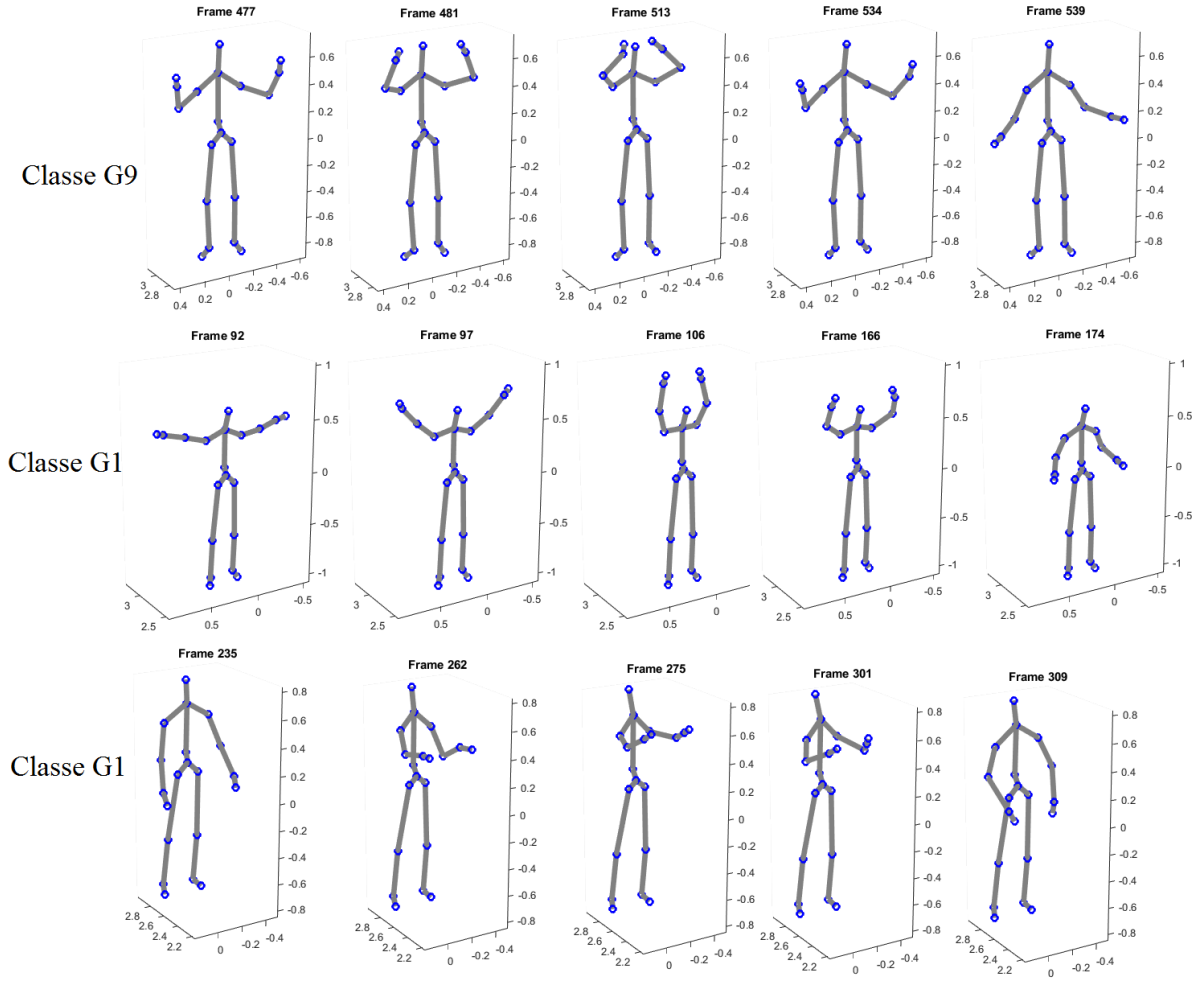

FIGURE 4.11 – OAD, MSRC-12 : Illustration de deux cas d'erreur de détection. La première ligne contient des frames d'une classe G9. Les deuxième et troisième rangées contiennent des frames de la même classe G1. Dans chaque rangée, la frame du milieu correspond au point d'action.

formances de détection, à savoir **4.00%**. En particulier, ceci suggère que les fenêtres curvilignes permettent d'adresser plus efficacement le problème des variations des vitesses d'exécution que lorsque plusieurs fenêtres temporelles sont employées, comme proposé dans [STHES15]. D'autre part, nous notons que la variante temporelle de **CuDi3D** est néanmoins meilleure que les approches temporelles précédentes, ce qui suggère que le processus de décision proposé, incluant l'utilisation de plusieurs classifieurs spécialisés, est également un composant important et une contribution de ce travail.

Dans la dernière expérience menée sur l'ensemble de données MSRC-12, nous proposons de mesurer l'impact de deux facteurs sur la performance de l'approche : le nombre de classifieurs curvilignes et leurs tailles curvilignes. Pour étudier l'impact du premier

Méthodes	CuDi3D Temporelle	**CuDi3D**
Video + Text	0.8220 ± 0.0800	**0.8540** ± 0.0670
Images + Text	0.7130 ± 0.1010	**0.7530** ± 0.0880
Text	**0.6790** ± 0.1240	0.6730 ± 0.1020
Video	0.7730 ± 0.0770	**0.8450** ± 0.0790
Images	0.6640 ± 0.1320	**0.7310** ± 0.1220
Total	0.7300	**0.7710**

TABLE 4.2 – OAD, MSRC-12 : Résultats d'une variante temporelle de **CuDi3D** sur la base de données MSRC-12. L'évaluation est menée dans le contexte OAD, suivant le protocole leave-subjects-out avec une latence de $\Delta = 333ms$.

facteur, nous avons conçu trois variantes de l'approche **CuDi3D** telles que : CuDi3D-Avg utilise un seul classifieur curviligne dont la taille est égale à la moyenne de toutes les tailles curvilignes, CuDi3D-Min utilise un seul classifieur curviligne dont la taille est égale au minimum de toutes les tailles curvilignes et le dernier CuDi3D-Three utilise trois classifieurs curvilignes dont les tailles sont respectivement le minimum, la moyenne et le maximum de toutes les tailles curvilignes. Pour toutes ces variantes, les autres composants restants sont inchangés. Les résultats selon un protocole leave-subjects-out sont rapportés dans la Table 4.3.

A partir de la Table 4.3, il apparaît d'abord que le choix des tailles curvilignes adé-

Méthodes	CuDi3D-Avg	CuDi3D-Min	CuDi3D-Three
Video + Text	0.6060 ± 0.1010	0.8590 ± 0.0540	**0.8620** ± 0.0520
Images + Text	0.5560 ± 0.0790	0.7160 ± 0.0590	**0.7460** ± 0.0750
Text	0.5570 ± 0.0770	0.6330 ± 0.1130	**0.6510** ± 0.0980
Video	0.5880 ± 0.0750	0.8060 ± 0.0740	**0.8140** ± 0.0740
Images	0.5350 ± 0.0980	0.7250 ± 0.1000	**0.7390** ± 0.0990
Overall	0.5680	0.7480	**0.7620**

TABLE 4.3 – OAD, MSRC-12 : Résultats expérimentaux de trois variantes de **CuDi3D** à savoir CuDi3D-Avg, CuDi3D-Min et CuDi3D-Three obtenus sur l'ensemble de données MSRC-12 selon le protocole leave-subjects-out à une latence de $\Delta = 333ms$.

quates est un point crucial. Par exemple, alors que les deux approches CuDi3D-Avg et CuDi3D-Min utilisent un seul classifieur, il existe une grande différence entre leurs performances. La performance inférieure de l'approche CuDi3D-Avg est due au fait que le classifieur utilisé n'est pas lancé si le déplacement curviligne cumulé ne dépasse pas la taille curviligne moyenne. Étant donné que de nombreuses actions ont des tailles curvilignes en-dessous de la taille moyenne, ces actions ne sont pas détectées, ce qui engendre des performances médiocres. Au contraire, l'approche CuDi3D-Min utilise la fenêtre curviligne la plus courte et même si elle repose sur un seul classifieur, elle permet d'obtenir de hautes performances. Cela confirme l'efficacité du jeu de descripteurs utilisé (**HIF3D**).

En outre, les résultats obtenus avec l'approche CuDi3D-Three montrent que l'augmentation du nombre de classifieurs de un à trois améliore les performances globales, mais reste inférieure à l'approche originelle **CuDi3D** composée de tous les classifieurs curvilignes possibles. Ceci démontre que l'utilisation de plusieurs classifieurs curvilignes de différentes tailles curvilignes permet de mieux gérer les variabilités spatiales. Nous pensons également que la différence entre les performances obtenues par les différentes variantes et celle réalisée par l'approche **CuDi3D** serait plus importante si les variabilités spatiales étaient plus prononcées dans la base de données MSRC-12.

Concernant le second facteur impactant la performance de l'approche proposée, à savoir les tailles curvilignes, nous présentons dans la Figure 4.12 les résultats de notre approche lorsque la taille curviligne de tous les classifieurs varie de 10 à 100% de leur valeurs initiales (100% correspond à l'approche originelle **CuDi3D**). Comme expliqué dans la section 4.2.2.2, les tailles curvilignes originelles de chaque classifieur sont déterminées en faisant la moyenne des déplacements curvilignes des échantillons appartenant à la classe d'action associée.

A partir de la Figure 4.12, nous pouvons conclure que la performance globale est améliorée quand les tailles curvilignes sont augmentées et se rapprochent des tailles utilisées dans notre approche (c'est-à-dire 100%). Ceci est cohérent avec le fait que les tailles curvilignes originelles sont calculées entre le début et le point d'action. En fait, réduire ces tailles augmente le risque de confusion des actions étant donné que les classifieurs commencent à prédire avant le moment où l'action est identifiable.

L'approche proposée est non seulement capable d'effectuer une détection d'actions en-ligne efficace, mais aussi de le faire en temps réel. Deux paramètres affectent ce temps de calcul, à savoir le nombre de classifieurs et les niveaux sur lesquels les descripteurs sont extraits. Sur la base de données MSRC-12, nous avons utilisé deux niveaux d'extraction et douze classifieurs. Le temps de calcul moyen par frame pour notre implémentation C++ est égal à environ 1.5 ms par frame. Le temps de calcul est mesuré sur une machine équipée d'un processeur Intel Core-i7 quad-core ayant une fréquence de 2.6 GHz et une

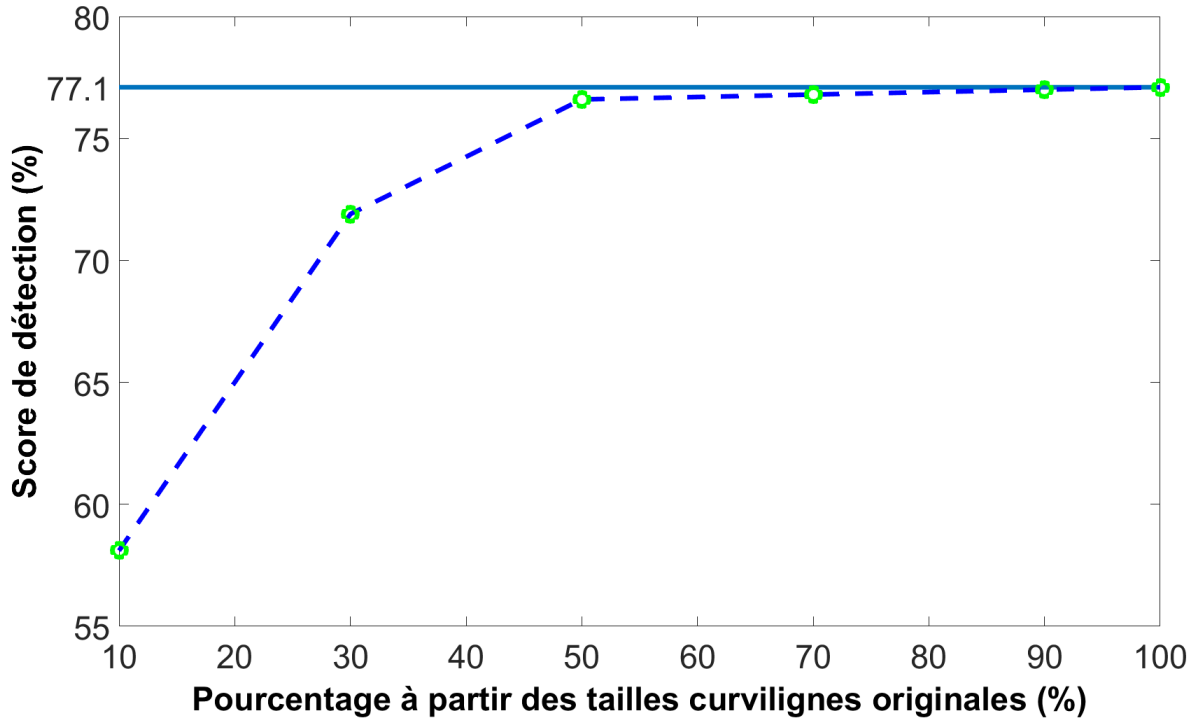

FIGURE 4.12 – OAD, MSRC-12 : Variation de la performance globale obtenue avec l'approche **CuDi3D** en fonction du pourcentage des tailles curvilignes.

mémoire RAM de 16 Go. Afin d'avoir un ordre de grandeur, nous fournissons aussi le temps de calcul de certaines approches de l'état de l'art dans la Table 4.4 sur cette même base.

ELS [MHT16]	RTMS [STHES15]	**CuDi3D**
10.7 ms	2.7 ms	1.5 ms

TABLE 4.4 – OAD, MSRC-12 : Comparaison des temps de calcul moyens par frame en millisecondes pour différentes approches OAD squelettiques. Voir la Table 4.1 pour les acronymes.

4.4.1.2 Base de données G3D

D'autres expériences ont été menées sur la base de données G3D. Cette base a été collectée pour permettre le développement de nouveaux algorithmes de détection et de reconnaissance d'actions pour le domaine des jeux vidéo [BMA12]. C'est un ensemble de 20 actions de jeu effectuées par 10 sujets qui ont reçu des instructions de base sur la

manière d'effectuer l'action. Les actions collectées sont regroupées en sept catégories : combat, golf, tennis, bowling, FPS, conduite et actions diverses. La plupart des séquences contiennent plusieurs actions dans un environnement intérieur contrôlé avec une caméra fixe. Chaque séquence est répétée trois fois par chaque sujet.

Pour permettre une comparaison avec les approches de l'état de l'art, nous avons retenu les séquences de la catégorie «Combat». Cette catégorie contient 5 classes : coup de pied à gauche, coup de pied à droite, coup de poing à gauche, coup de poing à droite et position de défense. Nous illustrons certaines de ces classes d'actions dans la Figure 4.13.

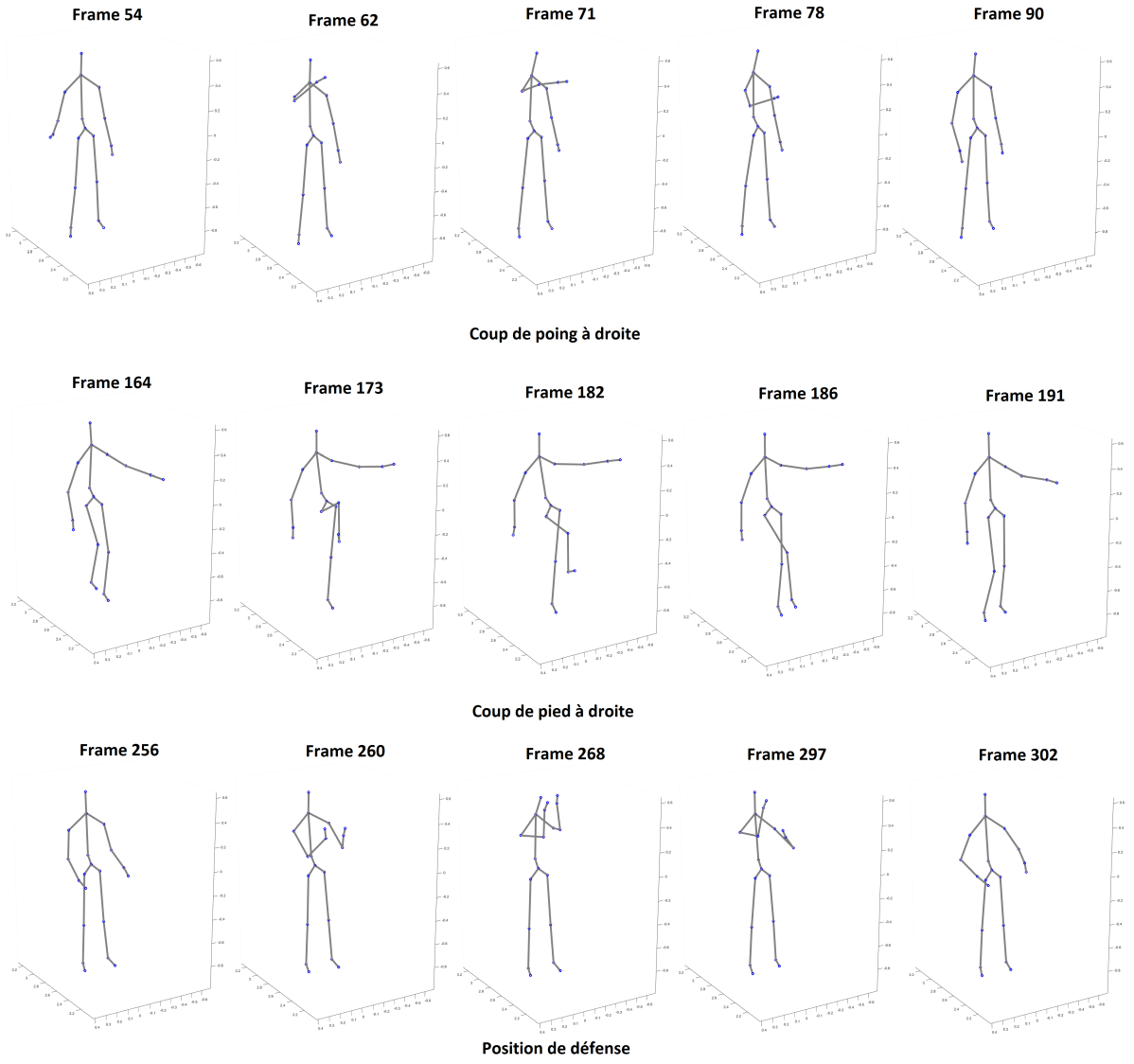

FIGURE 4.13 – OAD, G3D : Illustration de classes d'actions "coup de poing à droite", "coup de pied à droite" et "position de défense" appartenant à la base G3D.

Comme dans [BAM13, STHES15, BKK17, BMA14], nous avons d'abord suivi un protocole leave-subjects-out, dans lequel les données d'un des sujets sont retirées pour obtenir l'ensemble de test, tandis que le reste des données est utilisé pour construire notre modèle. Nous avons rapporté un score moyen pour 10 itérations. La Table 4.5 contient les scores obtenus.

Méthodes	DFS [BAM13]	RTMS [STHES15]	RF + ST [BKK17]	CAM [BMA14]	**CuDi3D**
F_score	0.9190	0.9210	0.9480	0.9780	**0.9890**

TABLE 4.5 – OAD, G3D : Résultats de la détection d'actions en-ligne sur la catégorie *Combat* de la base G3D selon le protocole leave-subjects-out. DFS = Dynamic Feature Selection ; RTMS = Real-Time Multi-Scale ; RF + ST = Random Forests using Spatio-Temporal Contexts ; CAM = Clustered Action Manifolds.

Comme indiqué dans la Table 4.5, l'approche proposée surpasse les méthodes de l'état de l'art et atteint le score de **98.90%** selon le protocole leave-subjects-out. Il est à noter que l'ensemble des résultats obtenus sur cette base de données sont particulièrement élevés. Ceci est, entre autre, dû au fait que chaque classe d'action active différentes parties du corps ce qui induit une très faible *similarité inter-classes*.

Une deuxième expérience a ensuite été menée sur la base G3D. Comme proposé par [BMA12], nous partageons la base de telle sorte que les données des 5 premiers sujets ont été utilisées pour l'apprentissage et celles des 5 sujets restants pour le test. Les résultats sont présentés dans la Table 4.6.

Ces résultats sont en accord avec les expériences précédentes. Plus précisément, l'approche proposée **CuDi3D** fait mieux que les meilleures méthodes de l'état de l'art avec un score de détection de **98.70%** suivant ce protocole. En fait, par rapport aux approches de l'état de l'art utilisant des fenêtres glissantes temporelles à taille fixe, telles que celles

Methods	AdaBoost [BMA12]	S-SW [LLX+16]	R-SW [LLX+16]	CA-RNN [LLX+16]	JCR-RNN [LLX+16]	**CuDi3D**
F_score	0.5850	0.7670	0.8330	0.9400	0.9620	**0.9870**

TABLE 4.6 – OAD, G3D : Résultats de la détection d'actions en-ligne sur la catégorie Combat de la base G3D selon le protocole de partage fixe proposé dans [BMA12]. S-SW = Support Vector Machine with Sliding Window ; R-SW = Recurrent Neural Network with Sliding Window ; CA-RNN = Classification Alone Recurrent Neural Network ; JCR-RNN = Joint Classification-Regression Recurrent Neural Network.

proposées dans [BMA12, LLX+16], les résultats obtenus confirment la robustesse de notre approche pour adresser les *variabilités temporelles*. Ces résultats sont d'autant plus importants qu'ils surpassent les récents systèmes plus complexes tels que ceux basés sur les réseaux de neurones récurrents présentés dans [LLX+16].

4.4.1.3 Base de données MAD

Nous évaluons enfin l'approche **CuDi3D** sur la base MAD contenant des actions multimodales [HYWDLT14]. MAD contient 40 longues séquences effectuées par 20 sujets (2 séquences par sujet), où chacun réalise 35 actions en continu dans chaque séquence. La longueur de chaque séquence est d'environ 2-4 minutes (4000-7000 frames). Les données squelettiques de 20 articulations sont fournies avec des vidéos RGB (240 × 320) et des images de profondeur 3D (240 × 320). Cependant, seules les données squelettiques sont utilisées dans nos expériences. Cette base de données est uniquement annotée avec les instants de début/fin. Nous donnons quelques exemples d'actions de cette base dans la Figure 4.14.

Pour être homogène avec les approches de l'état de l'art, une validation croisée de cinq folds sur les 20 sujets est utilisée comme protocole d'évaluation (4 sujets par fold). À chaque itération, les séquences étiquetées de quatre folds sont utilisées pour apprendre les tailles de déplacement curviligne par classe et pour former les différents classifieurs curvilignes. Les séquences du fold restant sont utilisées pour le test. Par exemple, dans la première validation croisée, les séquences des sujets 1 à 4 sont utilisées pour le test ($4 \times 2 = 8$ séquences), et les séquences des sujets 5 à 20 sont utilisées pour l'apprentissage ($16 \times 2 = 32$ séquences).

L'approche **CuDi3D** est évaluée à l'aide des métriques de précision et de rappel (cf. équation 4.12). Une action est considérée comme correctement détectée si la détection recouvre 50% de l'action. Nous calculons donc ces deux mesures et aussi le F_{score} pour permettre une comparaison avec les méthodes suivantes : MSO-SVM [HYWDLT14], SM-MED [HYWDLT14], Naive Bayes (NB) [EMS16] et Motion Segments (MS) [DBP+17]. Les résultats moyens sur les cinq folds sont reportés dans la Table 4.7.

Tout d'abord, les résultats montrent l'efficacité de notre approche par rapport aux méthodes OAD squelettiques de l'état de l'art sur la base de données MAD. En particulier, **CuDi3D** réalise le score de **0.8360**, ce qui correspond à une amélioration de +7.90%. Ceci est d'autant plus important que le protocole de test vise à la fois à détecter et à segmenter les séquences d'entrée. De plus, il s'agit d'une base composée de nombreuses actions (35 classes) dont certaines sont très similaires. Comparée à la méthode la plus efficace de Devanne et al. [DBP+17], qui décompose une action en segments temporels représentant des mouvements élémentaires, **CuDi3D** est plus efficace en termes de préci-

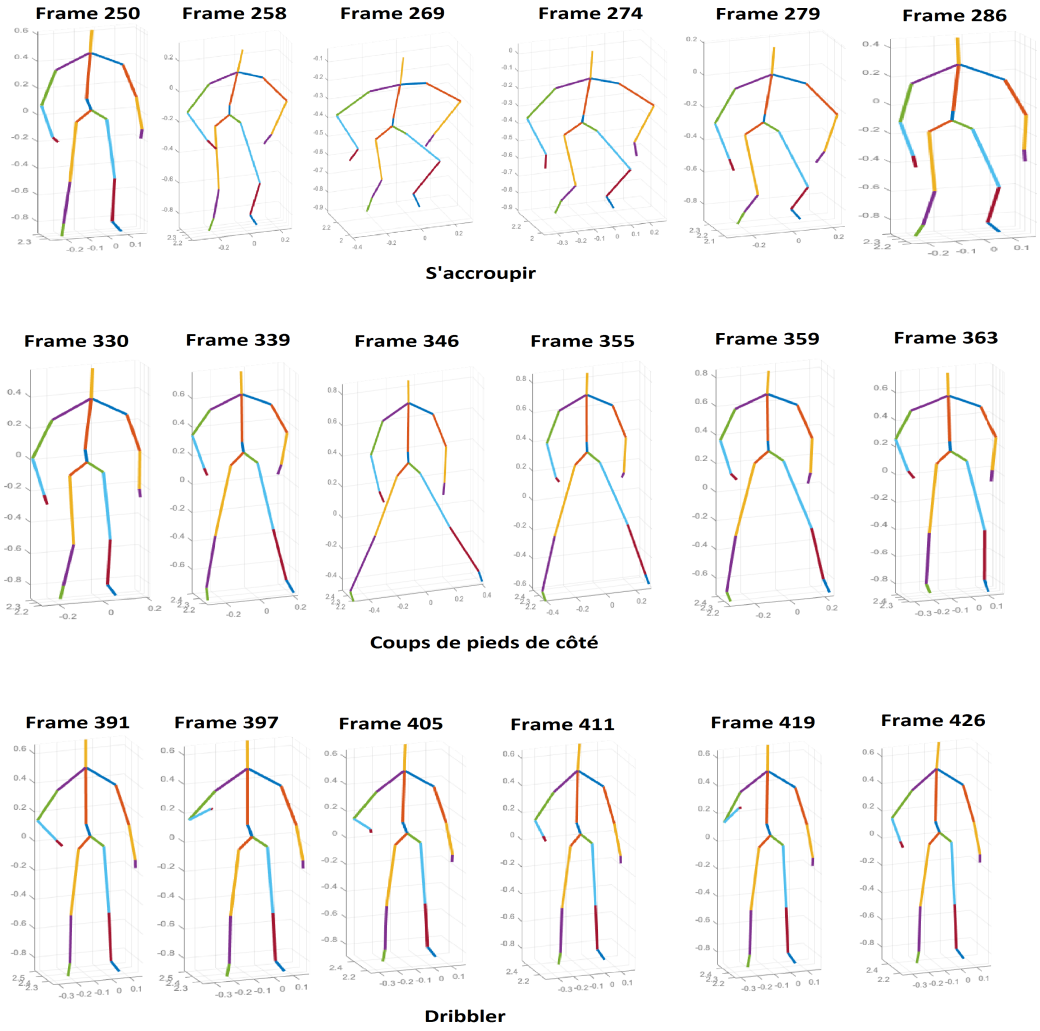

FIGURE 4.14 – OAD, MAD : Illustration des classes d'actions "s'accroupir", "coups de pieds de côté" et "dribbler" appartenant à la base MAD.

sion et de rappel. D'une part, le score plus élevé de précision reflète un faible nombre de fausses détections. Ceci est induit par le nouveau concept de fenêtre curviligne qui permet de considérer uniquement les segments de séquence où le mouvement a effectivement lieu, réduisant ainsi les faux positifs. D'autre part, le score plus élevé du rappel, correspondant à un ratio élevé de bonnes détections, est attribuable à deux composants de notre approche. Il s'agit d'une part du processus de décision qui combine, de manière appropriée, la décision locale de tous les classifieurs et d'autre part, de la procédure automatique de détermination des seuils, qui sont ajustés pour chaque classifieur et pour chaque classe.

La Figure 4.15 montre les résultats de détection sur la première séquence effectuée par le premier sujet de test. En plus de détecter correctement la plupart des actions

Méthodes	MSO-SVM [HYWDLT14]	SMMED [HYWDLT14]	NB [EMS16]	MS [DBP+17]	**CuDi3D**
Précision	0.2860	0.5740	0.7610	0.7210	**0.8520**
Rappel	0.5140	0.5920	0.7360	0.7970	**0.8200**
F_score	0.3680	0.5830	0.7480	0.7570	**0.8360**

TABLE 4.7 – OAD, MAD : Comparaison de l'approche **CuDi3D** avec de précédentes méthodes sur la base de données MAD suivant le protocole à cinq folds. MSO-SVM = Multiclass Structured Output SVM ; SMMED = Sequential Max-Margin Event Detectors ; NB = Naive Bayes ; MS = Motion Segments.

dans la séquence, l'approche proposée réussit à spécifier des instants de début et de fin pour les actions détectées qui sont très proches des annotations réelles. Cela résulte de la procédure de segmentation basée sur les scores de confiance et des tailles curvilignes des fenêtres comme expliqué dans la section 4.2.2.3. Dans cette procédure, nous considérons en fait que l'instant de fin de l'action (ou son point d'action) correspond à l'instant final de détection de cette action par le classifieur "gagnant". Le début d'action correspond à l'instant du début du cumul du score de la classe auquel on soustrait temporellement la taille de la fenêtre curviligne du classifieur. Cependant, on peut remarquer que pour

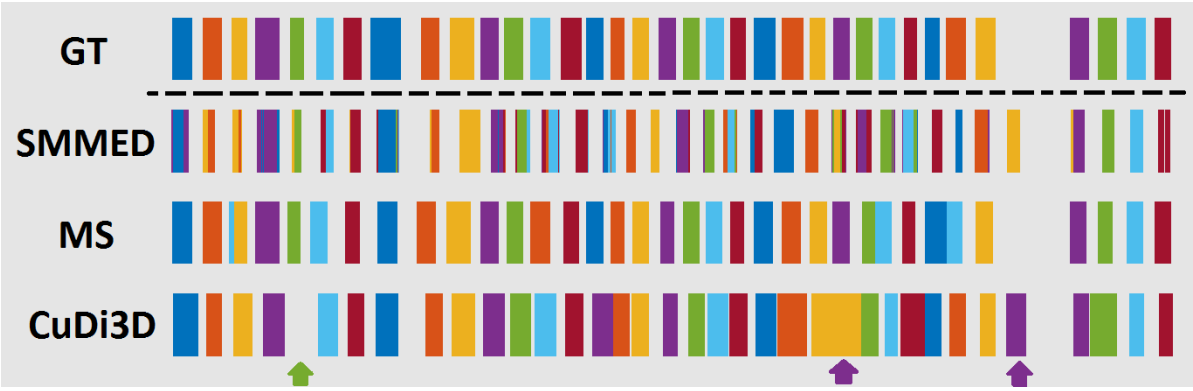

FIGURE 4.15 – OAD, MAD : Résultats de la détection d'actions (séquence-1 de sujet-1) pour la méthode SMMED [HYWDLT14] (deuxième rangée), la méthode MS (troisième rangée) et notre méthode **CuDi3D** (quatrième rangée) par rapport aux annotations de la vérité terrain (première rangée). Chaque classe a une couleur spécifique.

cette séquence, notre approche a raté deux détections (les deux premières flèches dans la Figure 4.15) et a émis un faux positif à la fin de la séquence (la troisième flèche dans la Figure 4.15). Après une visualisation de la séquence, il apparaît que pour le premier faux négatif aucun classifieur n'a cumulé assez de distance curviligne pour enclencher la détection, alors que pour le second faux négatif il y avait une forte confusion entre deux classes ce qui a retardé la décision finale. Enfin, le faux positif n'en ait pas un en réalité car il s'agit d'une erreur d'annotation de la séquence où le sujet effectue deux fois la même action (même couleur dans la figure) mais un seul label est rapporté dans les annotations de la vérité terrain.

4.4.2 Résultats de la reconnaissance d'actions pré-segmentées

Nous évaluons ici l'approche **CuDi3D** dans le contexte de la reconnaissance d'actions **pré-segmentées**. Nous rappelons que dans ce cas, comme pour l'évaluation menée dans le chapitre 3, chaque séquence à classifier comporte une seule action. Bien que le principal objectif de notre approche soit de résoudre le problème de la OAD, cette expérience vise à se faire une idée des performances de notre approche par rapport à celles de l'état de l'art, en particulier les méthodes basées sur l'apprentissage profond, pour modéliser et reconnaître des actions squelettiques pré-segmentées. Cette expérience est réalisée en utilisant la base de données HDM05 [MRC+07]. Cette base a déjà été utilisée pour la reconnaissance d'actions comme présenté dans le chapitre 3. Néanmoins, comme expliqué ci-après, nous allons utiliser une partie plus conséquente de la base. Nous présentons aussi dans cette section les résultats obtenus sur des actions pré-segmentées de la base MSRC-12 [FMKN12].

La base HDM05-Mocap [MRC+07] a déjà été introduite dans le chapitre 3. Pour cette expérimentation, nous adoptons le protocole d'évaluation proposé par [CC14] (sélectionner 65 classes, mener une validation croisée de 10 folds, où chaque fold contient 234 séquences appartenant aux différentes classes). Un prétraitement amorphologique est d'abord appliqué et les descripteurs **HIF3D** sont ensuite extraits selon un partitionnement temporel à deux niveaux (Niveau = 2). Nous entraînons 66 classifieurs SVM [CL11b] (65 classifieurs curvilignes + 1 classifieur global). Chacun est en mesure de reconnaître toutes les classes, mais un seul d'entre eux (celui avec la plus grande taille curviligne) est lancé à chaque frame. Chaque classifieur curviligne est basé sur une fenêtre glissante curviligne dont la taille correspond à l'une des 65 classes. Le classifieur supplémentaire est entraîné sur la totalité des séquences indépendamment de la taille curviligne de chaque séquence. Nous avons optimisé les paramètres SVM au moyen d'une validation croisée sur l'ensemble d'apprentissage.

La Table 4.8 rapporte les résultats de reconnaissance obtenus. Dans l'ensemble, notre approche fait mieux que les récentes approches basées sur des réseaux récurrents [CC14, DWW15, ZLX+16] et obtient un score de **99.40%**. Tout d'abord, l'approche proposée obtient de meilleurs résultats que le perceptron multicouche proposé dans [CC14]. Cela suggère que l'utilisation de plusieurs classifieurs spécialisés assure une meilleure modélisation du mouvement que l'utilisation d'un seul modèle. Ceci est aussi vrai en comparant les résultats avec ceux obtenus par la représentation à base d'un seul classifieur, dénommé **HIF3D** dans la Table 4.8, telle que présentée dans le chapitre 3. Il est possible donc de conclure que même en pré-segmenté il n'est jamais certain de savoir où débute et finit vraiment une action. Deuxièmement, par rapport aux réseaux récurrents hiérarchiques, entraînés séparément sur cinq parties du squelette humain, il semble que considérer le squelette entier et diviser plutôt le mouvement en plusieurs segments curvilignes est susceptible de mieux modéliser le mouvement. Notre approche fait mieux que toutes les architectures à base de LSTM proposées dans [ZLX+16].

Approche	**Taux de reco. (%)**
HIF3D + SVM + Niveau = 2	**91.00**
Multi-layer Perceptron [CC14]	95.60
Hierarchical RNN [DWW15]	96.90
Deep LSTM [ZLX+16]	96.80
Deep LSTM + Co-occurrence [ZLX+16]	97.00
Deep LSTM + Simple Dropout [ZLX+16]	97.20
Deep LSTM + In-depth Dropout [ZLX+16]	97.30
Deep LSTM + Co-occurrence + In-depth Dropout [ZLX+16]	97.30
CuDi3D + SVMs + Niveau = 2	**99.40**

TABLE 4.8 – Reconnaissance, HDM05 : Comparaison des résultats de l'approche **CuDi3D** avec ceux obtenus par les approches de l'état de l'art sur la base de données HDM05-Mocap, selon le protocole proposé dans [CC14].

Cette performance est liée à deux spécificités de notre approche. Premièrement, le fait de traiter un mouvement de manière progressive, c'est-à-dire frame par frame, et de cumuler les scores le long de ce traitement, augmente la robustesse de la décision finale. Cette décision, qui n'est pas nécessairement la dernière classe d'action prédite, est en fait basée sur la détection multiple par plusieurs classifieurs à différents niveaux spatio-temporels. Deuxièmement, la recherche de relations temporelles à l'intérieur d'un mouvement en focalisant et en guidant cette recherche sur des sous-segments de tailles

curvilignes prédéfinies est particulièrement efficace. Les approches de [DWW15, ZLX+16] à base de LSTM, dont la recherche de telles relations est entièrement automatique, ne prennent pas en compte les spécificités des actions modélisées pour guider cette recherche.

Nous évaluons aussi notre approche sur les actions pré-segmentées de la base MSRC-12 [FMKN12]. Cette expérience permet de consolider les résultats précédents sur la base HDM05 et de comparer les performances de notre approche aux résultats rapportés dans [EMT+13, HTGES13]. Pour ce faire, nous menons une validation croisée en divisant la base MSRC-12 en 4 folds et en faisant abstraction des différentes modalités utilisées lors de la capture. Tous les résultats sont reportés dans la Table 4.9.

Approche	**Taux de reco. (%)**
Regression-based classifier [EMT+13]	88.70
Cov3DJ + SVM + Niveau = 1 [HTGES13]	89.60
Cov3DJ + SVM + Niveau = 2 [HTGES13]	90.90
Cov3DJ + SVM + Niveau = 3 [HTGES13]	91.20
CuDi3D + SVMs + Niveau = 2	**96.30**

TABLE 4.9 – Reconnaissance, MSRC-12 : Comparaison des résultats de l'approche **CuDi3D** avec ceux obtenus par les approches de l'état de l'art sur la base de données MSRC-12, selon une validation croisée à 4 folds.

Le taux de reconnaissance rapporté par Ellis et al. [EMT+13] est de 88.70%, tandis que la meilleure configuration de Hussein et al. [HTGES13] atteint un score de 91.20%. Notre approche permet d'atteindre un score significativement plus élevé, à savoir **96.30%**. En plus des conclusions déjà dressées lors de l'évaluation avec la base HDM05, il est possible de conclure que modéliser une action en extrayant des descripteurs bien conçus est préférable à une simple comparaison entre les poses canoniques les plus discriminantes comme suggéré par Ellis et al. [EMT+13]. En effet, les descripteurs sont en mesure de faire ressortir des informations discriminantes souvent cachées dans une pose ou un ensemble de poses et non accessibles par une simple comparaison de ces poses. En outre, les résultats montrent également qu'il est préférable d'utiliser un ensemble réduit de descripteurs mais avec plusieurs classifieurs. Ceci permet de traiter une séquence à différents niveaux spatio-temporels, plutôt que d'extraire un grand nombre de descripteurs mais sur un nombre réduit de segments comme fait par Hussein et al. [HTGES13].

4.4.3 Résultats de la détection précoce

Nous présentons dans cette section les résultats de deux expérimentations menées sur la base MSRC-12 [FMKN12] afin d'évaluer les deux systèmes proposés pour faire de la détection précoce. Il existe peu de travaux ayant considéré la détection précoce d'actions dans un flot non segmenté de données squelettiques. La plupart des travaux sur la détection précoce ont été effectués en utilisant des vidéos RGB sur des actions pré-segmentées [Ryo11, CBB+13, DT06, LF12, LCS14] et, par conséquent, une comparaison directe n'est pas possible. Néanmoins, nous avons trouvé une approche proposée par Bloom et al. [BAM17] qui s'est intéressée à cette problématique. Pour permettre une comparaison avec les résultats présentés dans [BAM17], nous utilisons les données de la modalité "vidéo + texte" en suivant le même protocole de validation croisée à 10 folds. De plus, seules les actions iconiques sont considérées pour ces évaluations, à savoir : s'accroupir (G2), placer des lunettes (G4), tirer (G6), lancer (G8), changer d'arme (G10) et donner un coup de pied (G12).

La première expérimentation est conduite suivant le protocole d'évaluation que nous avons qualifié précédemment de détection précoce simplifiée. Il s'agit en fait du protocole le moins contraint où l'évaluation est conduite frame par frame indépendamment. Pour ce premier cas, nous nous basons sur la même métrique initialement proposée par Lan et al. [LCS14] qui utilisent la distance temporelle (en nombre de frames) pour rapporter les performances. Cette métrique consiste en fait à calculer les performances pour chaque frame située, au plus, à -20 frames du point d'action de chaque instance de test. Une moyenne est ainsi donnée pour chaque frame située dans l'intervalle [Point d'action - 20 frames, Point d'action].

Les approches auxquelles nous nous comparons sont : Clustered Spatio-Temporal Manifolds, Random Forests, AdaBoost, Dynamic Feature Selection. Ce sont en réalité des approches OAD qui ont été adaptées par [BAM17] pour des fins de détection précoce. En fait, avant l'étape de détection finale, ces approches classifient d'abord chaque frame, ce qui permet de considérer ces résultats pour une évaluation en détection précoce. Les résultats de l'approche **E-CuDi3D** et ceux obtenus par les approches de [BAM17] sont rapportés dans la Figure 4.16. Nous rapportons aussi dans cette figure les résultats des approches **CuDi3D-10** et **CuDi3D-100** utilisant un seul modèle ayant respectivement une taille de 10% et 100%.

Sur la base des résultats de la Figure 4.16, il est possible de voir que l'approche à trois modèles est globalement meilleure que la plus performante des approches proposées par [BAM17]. En particulier, notre approche se distingue pour les instants les plus éloignés du point d'action. Ceci suggère notamment que, du point de vue de la précocité, notre

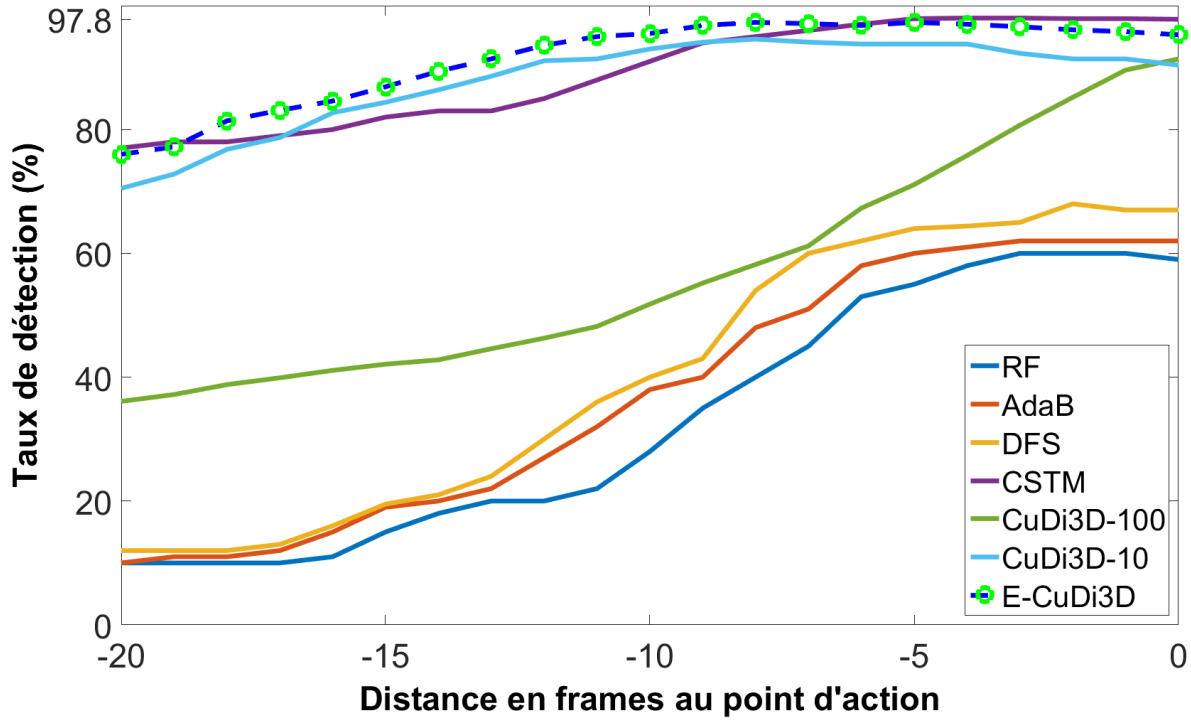

FIGURE 4.16 – Détection précoce simplifiée, MSRC-12 : Résultats obtenus pour 20 frames avant les points d'actions sur la base MSRC-12 en détection précoce simplifiée. L'évaluation est menée sur six classes d'actions suivant le protocole leave-subjects-out. E-CuDi3D = notre approche ; CuDi3D-10 et CuDi3D-100 sont les modèles à 10% et 100% de tailles curvilignes ; CSTM = Clustered Spatio-Temporal Manifolds [BAM17] ; RF = Random Forests [BAM17] ; AdaB = AdaBoost [BAM17], DFS = Dynamic Feature Selection [BAM17].

approche est plus intéressante alors que l'approche CSTM de [BAM17] prend légèrement le dessus en se rapprochant des points d'actions. Cela caractérise la robustesse de la notion de fenêtres curvilignes qui permettent d'adresser les problèmes de variabilités temporelles et la puissance de représentation des descripteurs **HIF3D**. En outre, par rapport aux deux courbes, traduisant les performances des deux approches de **CuDi3D-10** et **CuDi3D-100**, l'approche **E-CuDi3D** est supérieure et permet en effet de combiner les avantages de chacune d'elles.

Dans la deuxième expérimentation, nous évaluons l'approche **E-CuDi3D** dans le cadre plus complexe de la détection précoce au sens donné par le premier protocole. Suivant ce protocole, dès qu'une erreur est produite elle entraîne une erreur sur l'action. Comme pour le protocole précédent, une validation croisée est réalisée. Néanmoins lors de l'évaluation, nous calculons la mesure F_score qui combine précision et rappel et dans laquelle les faux positifs et les faux négatifs sont comptabilisés (cf. équation 4.12). Les résultats sont

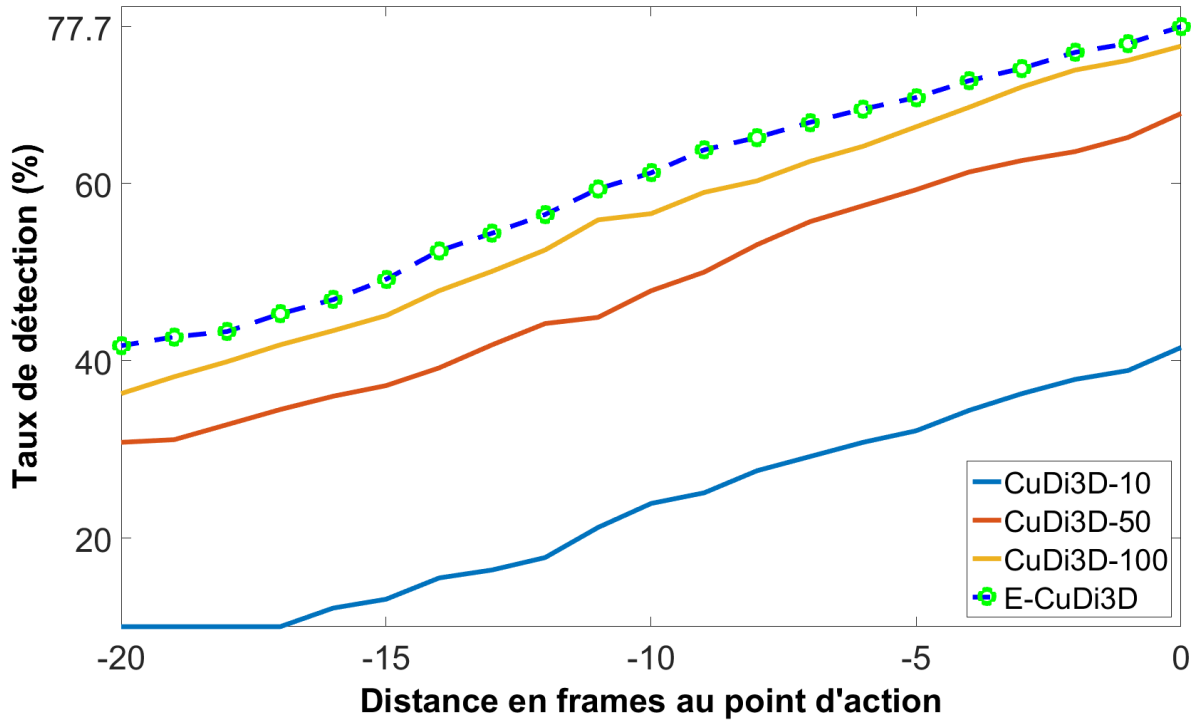

FIGURE 4.17 – Détection précoce, MSRC-12 : Résultats obtenus pour 20 frames avant les points d'action sur la base MSRC-12 en détection précoce.

donnés sous forme de courbe dans la Figure 4.17.

Vu qu'aucune approche n'a considéré auparavant ce contexte d'évaluation, nous avons rapporté non seulement les résultats de notre approche (**E-CuDi3D**) mais aussi ceux des trois modèles séparément, à savoir **CuDi3D-10**, **CuDi3D-50** et **CuDi3D-100**. Nous relevons alors que l'approche **E-CuDi3D** permet de combiner les avantages de chacun des trois modèles et réussit à atteindre des scores intéressants bien avant l'instant du point d'action. Ces résultats peuvent servir de référence pour de futurs travaux considérant ce cadre d'évaluation plus complexe mais aussi plus proche d'une exploitation réaliste de la détection précoce.

4.5 Conclusion

Dans ce chapitre, nous avons considéré la problématique de détection en-ligne d'actions squelettiques dans un flot non segmenté. Il s'agit d'une problématique ayant plus d'intérêt pratique que la reconnaissance d'actions pré-segmentées, mais elle est plus complexe que cette dernière. Dans le cadre de notre étude, nous avons d'abord spécifié trois types de difficultés qu'une approche de la OAD devrait considérer, à savoir les *variabilités temporelles*, les *variabilités spatiales inter-classes* ainsi que les *variabilités spatiales intra-classe*. Nous avons ensuite proposé une approche originale de la OAD, dénommée **CuDi3D**, qui permet de mieux adresser ces difficultés.

En particulier, en ce qui concerne les *variabilités temporelles*, nous avons introduit le concept de fenêtre curviligne. Une telle fenêtre englobe des segments de trajectoire de longueur homogène et ce indépendamment du temps passé par un sujet à effectuer ce mouvement. Ceci est fondamentalement différent des approches de l'état de l'art dans la mesure où elles utilisent des fenêtres glissantes temporelles qui ne permettent d'extraire des descripteurs que sur des segments d'une durée pré-définie, et donc sont sensibles aux variations de vitesse. Ensuite, pour aborder les *variabilités spatiales inter-classes*, nous proposons de lancer plusieurs classifieurs curvilignes en parallèle pour analyser le flot d'entrée avec différentes fenêtres curvilignes. Enfin, nous adressons les *variabilités spatiales intra-classe* au moyen d'un système de fusion dans lequel les décisions locales et les scores de confiance sont combinés. Le but est de détecter les actions en cours le plus tôt possible tout en réduisant les confusions possibles entre les classes.

Cette nouvelle approche a été adaptée et étendue pour résoudre deux autres problèmes. En effet, d'une part nous avons proposé une nouvelle approche pour résoudre le problème de reconnaissance d'actions pré-segmentées, déjà abordé dans le chapitre 3. Cette approche se base sur une analyse frame par frame de la séquence d'action en utilisant à chaque fois le classifieur de plus grande taille curviligne possible. Les scores de confiance donnés par chaque classifieur sont cumulés dans un histogramme global, qui sert à la fin du traitement pour décider de la classe identifiée. D'autre part, nous avons proposé une autre approche pour adresser le problème de détection précoce d'actions non segmentées. Cette autre variante combine trois modèles de détection à base de fenêtres curvilignes, où chaque modèle utilise des tailles curvilignes différentes de manière à scruter le flot de données à court, moyen et long terme.

Nous avons d'abord évalué l'approche **CuDi3D** sur trois bases de données squelettiques dans le contexte de la détection en-ligne d'actions non segmentées. De meilleurs résultats ont été obtenus, attestant de la supériorité de cette approche par rapport aux autres approches de la littérature. Nous avons aussi isolé la contribution de chacun des

composants de l'approche et avons montré l'intérêt de chacun. En particulier, le fait de remplacer dans notre approche les fenêtres curvilignes par des fenêtres temporelles conventionnelles a détérioré les performances de l'approche. L'approche proposée est d'autant plus intéressante que les paramètres nécessaires, en particulier les seuils de confiance, sont automatiquement optimisés à partir des données d'apprentissage.

Nous avons ensuite montré que l'approche, ayant été adaptée pour la reconnaissance d'actions pré-segmentées, réalise de meilleures performances que celles obtenues par des approches de l'état de l'art, notamment les approches récentes à base d'apprentissage profond (*deep learning*). Enfin, nous avons obtenu des résultats prometteurs avec l'approche à trois modèles dans le contexte de la détection précoce suivant un protocole d'évaluation réaliste.

Chapitre 5

Applications

5.1 Introduction

Dans ce dernier chapitre, nous présentons trois travaux dans lesquels nous avons appliqué les approches de reconnaissance et de détection proposées dans les chapitres précédents. Tout d'abord, nous présentons une approche de reconnaissance de gestes dynamiques de la main ayant été conçue sur la base des descripteurs **HIF3D** (cf. chapitre 3). Dans un deuxième temps, nous présentons un système d'interaction sous forme d'un jeu, où un sujet utilise des gestes de la main pour contrôler et faire déplacer un humain virtuel. Dans la dernière section, nous discutons de l'utilisabilité de notre approche de détection et de la notion de distance curviligne pour améliorer l'animation d'humain virtuel en temps réel.

5.2 Reconnaissance de gestes dynamiques de la main

L'idée globale de cette section est d'évaluer notre représentation conçue pour des actions corps-complet, notamment les descripteurs **HIF3D**, pour modéliser des gestes dynamiques de la main. En effet, il existe aujourd'hui un nombre croissant d'approches s'intéressant à la reconnaissance des gestes effectués avec la main. La finalité visée, comme c'est le cas pour les actions corps-complet, est d'ouvrir de nouvelles modalités d'interaction gestuelles qui pourraient être plus intuitives que les modalités d'interaction classiques telles que le clavier et la souris. Plusieurs domaines d'application se prêtent à de telles fonctionnalités dont la réalité virtuelle et augmentée, la reconnaissance du langage des signes, les jeux, la robotique, etc.

Les approches de l'état de l'art traitant de la reconnaissance des gestes de la main peuvent être regroupées en deux catégories principales : la reconnaissance des gestes sta-

tiques et la reconnaissance des gestes dynamiques. Les premières se concentrent uniquement sur la posture en extrayant les régions d'intérêt ou les silhouettes des mains. Les secondes considèrent plutôt la progression des positions des articulations de la main dans le temps. Ces dernières permettent de considérer la dynamique du geste et sont alors plus adaptées aux interfaces d'interaction.

Nous considérons donc dans cette section la reconnaissance des gestes dynamiques de la main en se basant sur des données squelettiques. En particulier, nous proposons d'abord une nouvelle représentation basée sur les descripteurs **HIF3D** (cf. chapitre 3). De plus, nous introduisons une nouvelle base de gestes dynamiques de la main que nous avons collectée. Cette base permet de considérer cette problématique avec plus de contraintes, notamment en comportant des gestes effectués avec une ou deux mains.

5.2.1 Représentation des gestes dynamiques de la main

Au lieu de reconnaître un geste de la main en se basant uniquement sur l'information brute, c'est-à-dire les positions absolues des articulations ou leurs angles, nous proposons d'extraire les descripteurs **HIF3D** sur les trajectoires formées. Ces descripteurs permettent de récupérer des informations caractérisant les trajectoires des doigts des mains. En fait, malgré la différence des amplitudes des trajectoires produites d'une part par le corps-complet, et d'autre part uniquement avec les doigts des mains, le pattern est considéré comme le résultat d'un mouvement 3D produit par un humain. Nous proposons donc de faire un parallèle entre les doigts et les trajectoires des articulations du corps-complet afin d'exploiter la puissance de modélisation des descripteurs conçus précédemment pour les actions corps-complet. Ceci est particulièrement intéressant car les descripteurs **HIF3D** sont eux-mêmes le résultat d'un transfert opéré à partir de descripteurs 2D. Ce faisant, nous voulons aller au bout de la fusion des techniques permettant la modélisation de trajectoires créées par l'humain lors de la performance d'un geste qu'il soit 2D, 3D corps-complet ou 3D effectué avec la main.

En particulier, nous considérons comme données d'entrée les positions 3D brutes des extrémités du doigt ainsi que les positions de la paume et du poignet (Figure 5.1a). Les positions de chacune de ces articulations sont fournies dans un repère centré sur le dispositif de capture (par exemple le Leap Motion). Lors de l'exécution d'un geste donné, les positions successives de chaque articulation constituent une trajectoire 3D. A la différence des actions corps-complet, pour lesquelles une normalisation amorphologique est appliquée, les trajectoires de la main ainsi obtenues ne sont pas normalisées. Ces trajectoires sont ensuite assemblées pour former un seul motif 3D S composé de plusieurs trajectoires 3D : $S = \{s_1, ..., s_T, s_{T+1}, ..., s_{2T}, s_{2T+1}, ..., s_{3T}, s_{3T+1}, ..., s_n\}$, où T est la lon-

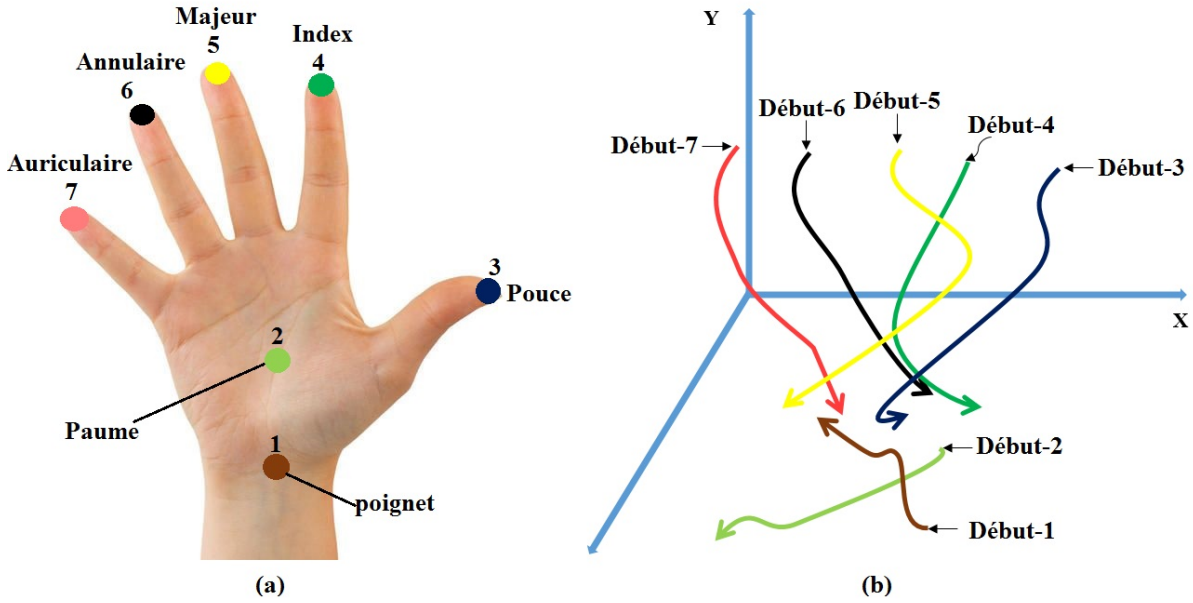

FIGURE 5.1 – (a) Articulations sélectionnées pour notre représentation du geste dynamique de la main. (b) Illustration d'un pattern 3D multistrokes résultant de l'ensemble des trajectoires des doigts.

gueur de chaque trajectoire unique et $n = K \times T$ est le nombre de points dans S. K est le nombre d'articulations considérées (K = 7 pour une base de données ne comportant que des gestes à une seule main ou K = 14 pour une base de données à deux mains). Chaque point $s_i = (x_i, y_i, z_i)$ est situé dans l'espace tridimensionnel.

Une illustration du motif 3D obtenu S est fournie dans la Figure 5.1b. Par exemple, les première et deuxième trajectoires correspondent respectivement à l'articulation du poignet et à l'articulation de la paume. Nous adoptons l'ordre suivant pour l'assemblage des trajectoires : poignet, paume, pouce, index, majeur, annulaire et auriculaire.

Le motif 3D ainsi obtenu est ensuite transmis à l'extracteur de descripteurs **HIF3D**. De plus, comme ces descripteurs permettent seulement d'exprimer la variation de la forme de la main, c'est-à-dire l'information spatiale, nous devons en plus capturer l'information temporelle. En fait, comme pour les actions corps-complet, certains gestes tels que les gestes inversés, ont les mêmes caractéristiques spatiales et ne peuvent être distingués que sur la base de leur information de synchronisation temporelle. Nous extrayons donc les descripteurs **HIF3D** selon une hiérarchie temporelle à deux niveaux comme présenté dans la Figure 5.2.

Ainsi, la combinaison de chaque fenêtre consiste en une concaténation de tous les descripteurs extraits sur l'ensemble de la séquence et sur les trois sous-séquences qui se

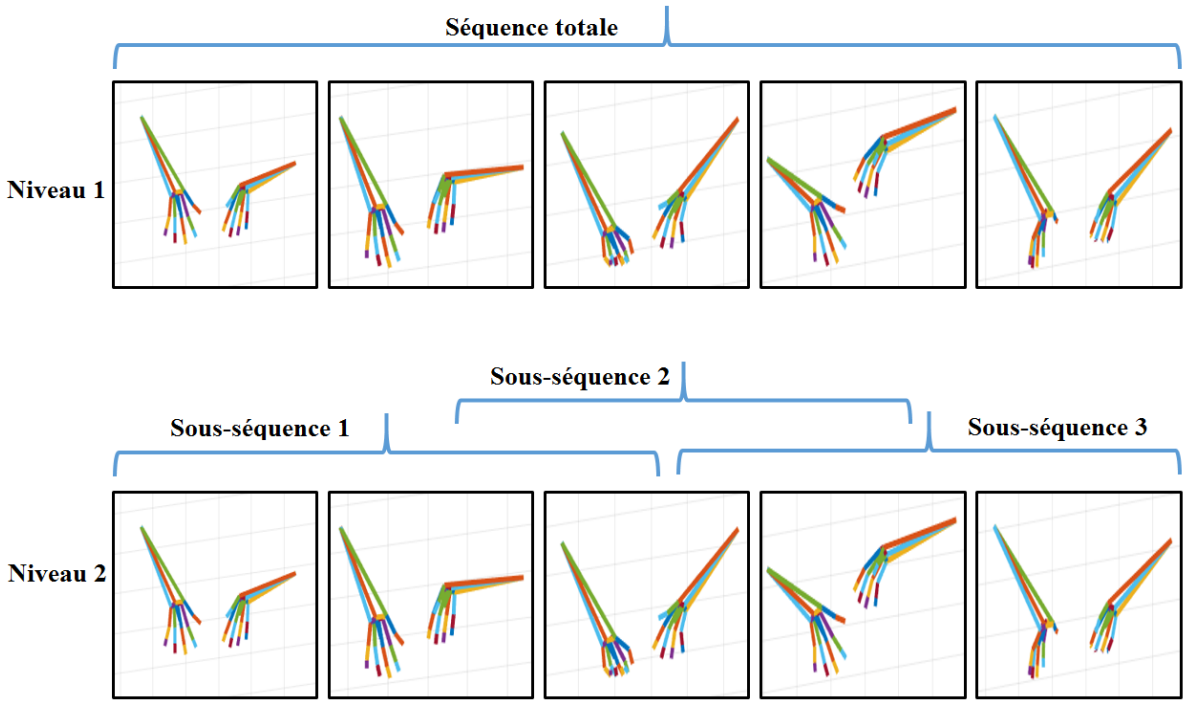

FIGURE 5.2 – Illustration des séquences considérées pour extraire notre représentation sur deux niveaux temporels.

chevauchent. Comme illustré dans la Figure 5.3, le pattern 3D considéré au niveau 1 est composé de toutes les trajectoires (3 trajectoires pour simplifier) dans un ordre déterminé alors que les patterns du niveau 2 sont formés par les segments issus du découpage temporel de ces trajectoires. Ainsi, la première et la deuxième sous-séquence du niveau 2 ne comporte qu'un petit segment de la première trajectoire, alors que la troisième sous-séquence comporte la majeure partie de cette première trajectoire (stroke). Au contraire, les trois sous-séquences comporte chacune des segments conséquents de la deuxième et

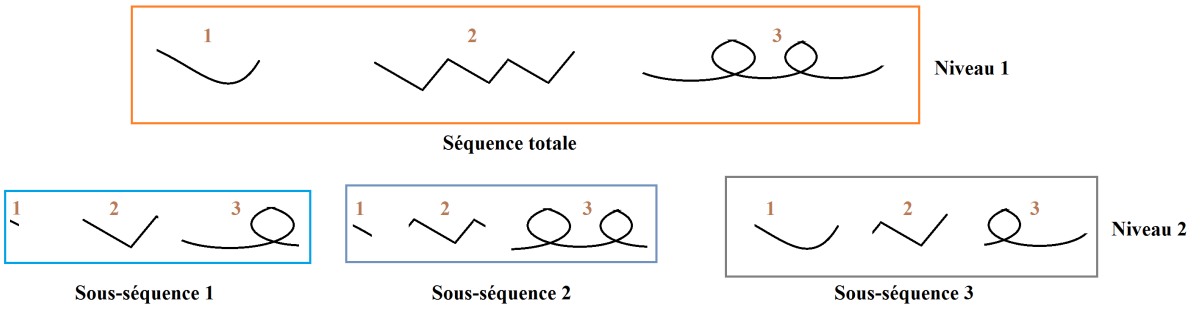

FIGURE 5.3 – Illustration avec trois trajectoires (strokes) des patterns 3D issus du découpage temporel à deux niveaux d'une séquence de geste de la main.

la troisième trajectoire. On en déduit que la première articulation ne s'est pas beaucoup déplacée au début et au milieu de l'action alors qu'elle se déplace principalement en fin d'action. En procédant de cette manière il est donc possible d'inclure la dépendance temporelle dans notre représentation.

Cela conduit à une représentation finale de seulement 356 descripteurs ($89\times1+89\times3$), qui représente une dimension très réduite par rapport aux représentations de l'état de l'art pouvant contenir des milliers de descripteurs. En ce qui concerne la classification, nous utilisons des machines à vecteurs de support (SVM). Ce classifieur est utilisé avec un noyau polynomial et des paramètres optimisés au moyen d'une grille de recherche. Nous utilisons pour celà la bibliothèque LIBSVM [CL11b].

5.2.2 Collection d'une nouvelle base de données des gestes dynamiques de la main : LMDHG

Afin de permettre des évaluations plus approfondies des systèmes de reconnaissance des gestes dynamiques de la main, nous avons procédé à la collecte d'une nouvelle base de gestes. En effet, nous avons remarqué que les bases de données existantes ne sont composées que de clips très courts (environ 30 frames) au cours desquels les gestes enregistrés sont réalisés avec une seule main. Nous avons également constaté que les gestes enregistrés sont parfaitement réalisés et que ces enregistrements ne comportent pas de bruit ou de distorsion comme c'est souvent le cas en utilisation réelle. Enfin, les bases de données précédentes ne contiennent que des gestes pré-segmentés ce qui empêche d'évaluer les méthodes de segmentation. La nouvelle base que nous proposons vise à combler ces lacunes. Ci-après, nous décrivons la base de données collectée, dénommée LMDHG pour *LeapMotion Dynamic Hand Gesture*[1], qui a été mise à la disposition de la communauté.

A l'opposé des bases de données de gestes dynamiques existantes, LMDHG contient des séquences non segmentées de gestes effectués soit avec une seule main soit avec les deux mains (un exemple est montré dans la Figure 5.4).

Il y avait 21 participants, chaque participant a effectué au moins une séquence, résultant en 50 séquences. Chaque séquence contient des instances de 13 classes conduisant à un total de 608 gestes. À la fin de chaque geste, on a demandé au participant de garder ses mains au-dessus du dispositif de capture (Leap Motion) avant d'effectuer un autre geste. Nous avons étiqueté ce non-geste comme une classe pause. De plus, l'ordre des classes dans chaque séquence est aléatoire.

Chaque frame contient les coordonnées 3D de 46 articulations (23 articulations pour chaque main). Si l'une des mains n'est pas suivie, la position de ses articulations est mise

1. https ://www-intuidoc.irisa.fr/en/english-leap-motion-dynamic-hand-gesture-lmdhg-database/

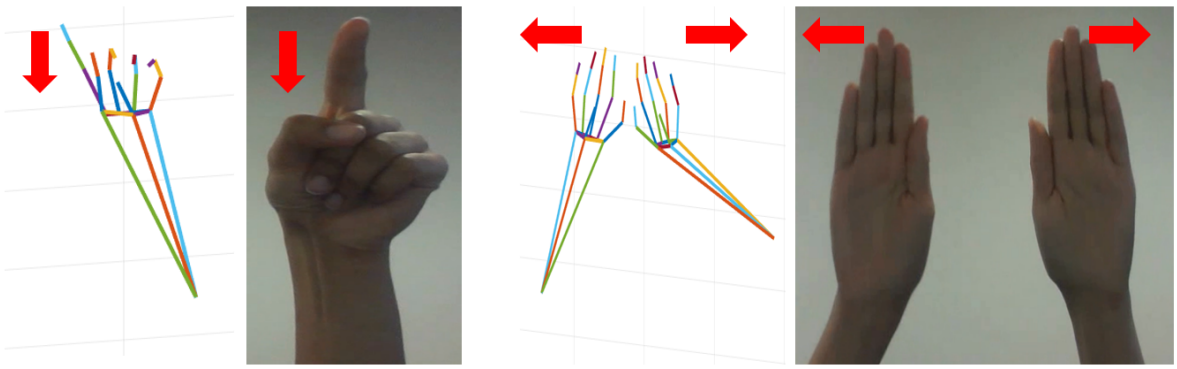

FIGURE 5.4 – (a) Illustration du tracé d'une ligne effectué avec une main. (b) Geste de zoom effectué avec deux mains.

à zéro. Nous fournissons également, dans la vérité terrain, les étiquettes et les instants de début et de fin de chaque geste dans chaque séquence.

Les classes de gestes composant notre base de données sont listées dans Table 5.1. Ces gestes sont choisis pour évaluer la robustesse des systèmes de reconnaissance et de détection face à deux aspects : (1) Évaluer la robustesse aux gestes effectués avec une ou deux mains ; (2) Évaluer la robustesse aux données bruitées/manquantes.

Classes de geste	**#Mains**	**Étiquette**
Point to	1	HG1
Catch	1	HG2
Shake with two hands	2	HG3
Catch with two hands	2	HG4
Shake down	1	HG5
Shake	1	HG6
Draw C	1	HG7
Point to with two hands	2	HG8
Zoom	2	HG9
Scroll	1	HG10
Draw Line	1	HG11
Slice	1	HG12
Rotate	1	HG13

TABLE 5.1 – Liste des classes de gestes de la base de données LMDHG.

5.2.3 Résultats expérimentaux et discussion

Dans cette section, nous évaluons d'abord notre approche de reconnaissance des gestes dynamiques de la main sur une base de données récemment publiée, à savoir DHG dataset [DSWV16]. Nous fournissons ensuite des résultats préliminaires sur notre propre base LMDHG en considérant uniquement le cadre de **reconnaissance avec des séquences pré-segmentées**.

5.2.3.1 Base de données DHG

DHG est une base de données de gestes dynamiques de la main, présentée dans [DSWV16]. Elle contient des instances appartenant à 14 classes de gestes, effectués de deux manières : en utilisant soit un doigt soit la main entière (illustration dans la Figure 5.5). Suivant ces deux manières, chaque geste est effectué entre 1 et 10 fois par 28 participants, résultant en 2800 instances. Chaque frame contient les coordonnées de 22 articulations à la fois dans l'espace de l'image 2D et dans l'espace 3D formant un squelette de la main complète. Une description des gestes de cette base est donnée dans la Table 5.2.

Différents protocoles peuvent être menés sur cette base. Nous retenons deux protocoles qui ne prennent en compte que les données squelettiques dans l'espace 3D. Suivant les deux protocoles, nous utilisons 1960 instances pour l'apprentissage du modèle de reconnaissance et 840 instances pour l'évaluation afin de permettre une comparaison équitable avec les méthodes de l'état de l'art.

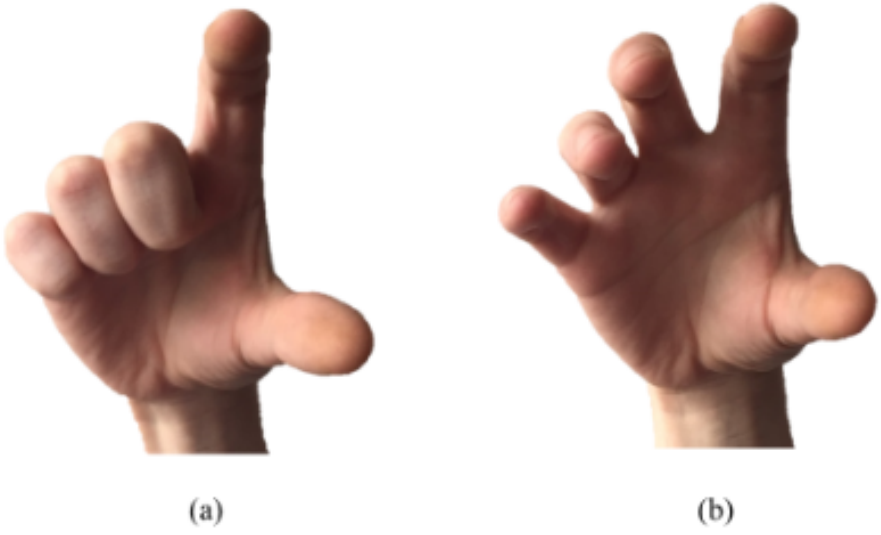

FIGURE 5.5 – Deux images d'une main illustrant le geste de "saisir" effectué **(a)** avec un doigt et **(b)** avec la main entière [DSWV16].

	Label des classes	Étiquette	Nombre de séquences par classe	Nombre de sujets
	Saisir (Grab)	G	200	20
	Étendre (Expand)	E	200	20
	Pincer (Pinch)	P	200	20
	Rotation sens horaire (Rotation CW)	R-CW	200	20
	Rotation sens antihoraire (Rotation CW)	R-CCW	200	20
	Appuyer (Tap)	T	20	10
	Défiler vers la droite (Swipe Right)	S-R	200	20
	Défiler vers la gauche (Swipe Left)	S-L	200	20
	Défiler vers le haut (Swipe Up)	S-U	200	20
	Défiler vers le bas (Swipe Down)	S-D	200	20
	Marquer un X (Swipe X)	S-X	200	20
	Marquer un V (Swipe V)	S-V	200	20
	Marquer un + (Swipe +)	S-+	200	20
	Secouer (Shake)	Sh	200	20
Total	14 classes	-	2800 séquences	20 sujets

TABLE 5.2 – Tableau récapitulatif des propriétés de la base DHG en termes de nature des actions, étiquette, nombre de classes d'actions, nombre de séquences et nombre total des sujets.

Selon le premier protocole, les gestes sont regroupés en 14 classes alors que suivant le deuxième protocole les même gestes sont regroupés en deux fois plus de classes à savoir 28 classes. Ceci est possible puisque tous les gestes sont effectués suivant deux manières (voir la description de la base plus haut). En fait, suivant le premier protocole, on considère deux instances données représentant le même geste (par exemple "attraper") mais où l'un est effectué avec un doigt et l'autre est effectué avec tous les doigts, comme appartenant à la même classe. Au contraire, suivant le deuxième protocole, de telles instances appartiennent à deux classes différentes. Le deuxième protocole est évidemment plus difficile car la variabilité inter-classes est plus faible que dans le premier protocole. La Table 5.3 rapporte les résultats de notre approche avec ceux des méthodes de l'état de l'art dans les cas de 14 et 28 gestes.

D'après ces résultats, notre approche surpasse les méthodes de l'état de l'art dans le cas où 14 gestes sont considérés, avec un score final de **90.48%**. De plus, nous obtenons un score de **80.48%** en considérant le cas plus difficile de 28 gestes. Dans ce dernier cas, nous obtenons le deuxième meilleur score, très proche de la performance de [DSWV16], qui est de 81.90%.

Plusieurs conclusions peuvent être tirées sur la base des résultats obtenus. Tout d'abord, nous avons montré que la reconnaissance des actions 3D et des gestes dynamiques 3D de la main partage des propriétés similaires car elles ont été traitées de manière identique. Les résultats témoignent également des mérites des descripteurs **HIF3D** dans la modélisation

Méthode	**14 gestes (%)**	**28 gestes (%)**
HoWR [DSWV16]	35.61	-
SoCJ [DSWV16]	63.29	-
HoHD [DSWV16]	67.64	-
Oreifej *et al.* [OL13, DSWV$^+$17]	78.53	74.03
Devanne *et al.* [DWB$^+$15, DSWV$^+$17]	79.61	62.00
SoCJ + HoHD [DSWV16]	82.29	-
Guerry *et al.* [DSWV$^+$17]	82.90	71.90
SoCJ + HoHD + HoWR [DSWV16]	83.07	80.00
Ohn-Bar *et al.* [OBT13, DSWV$^+$17]	83.85	76.53
De Smedt *et al.* [DSWV16, DSWV$^+$17]	88.24	**81.90**
Notre approche	**90.48**	80.48

TABLE 5.3 – Comparaison entre notre approche et les approches de l'état de l'art en considérant 14 et 28 gestes sur l'ensemble de données DHG.

des gestes de la main et de l'intérêt de considérer des descripteurs de haut niveau au lieu d'utiliser des données brutes ou des descripteurs de bas niveau. Enfin, il est important de noter que notre approche utilise un sous-ensemble de 7 articulations sur 22 et une

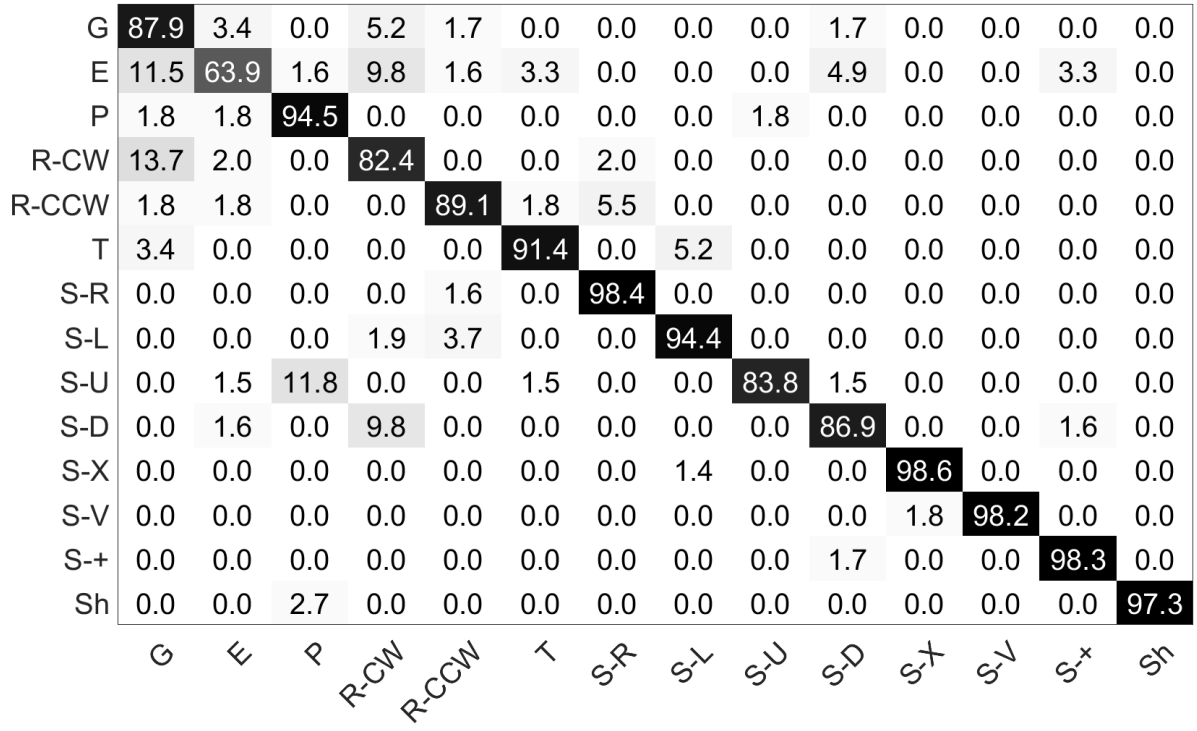

FIGURE 5.6 – Notre matrice de confusion sur la base DHG en utilisant 14 gestes.

perspective intéressante serait d'étudier l'impact de considérer plus d'articulations. Pour une vue détaillée de nos résultats lors de l'utilisation de 14 gestes, nous fournissons la matrice de confusion dans la Figure 5.6.

5.2.3.2 Base de données LMDHG

Dans cette section, le but est de fournir des résultats préliminaires sur la base de données maison collectée dans le contexte de la reconnaissance de gestes de la main. Nous avons utilisé 70% de la base pour entraîner le modèle, à savoir les séquences 1 à 35, tandis que 30% des séquences sont utilisées lors du test (séquences de 36 à 50). Selon cette répartition, les (sept) personnes figurant dans l'ensemble test n'ont pas participé à la collecte de l'ensemble d'apprentissage.

Le score global obtenu sur cette deuxième base est de **84,78%**. Ce score est particulièrement prometteur car la base de données LMDHG contient des gestes bruités et certains sont incomplets. De plus, comme l'ensemble de données contient des gestes effectués avec une seule main mais aussi des gestes nécessitant deux mains, le résultat témoigne également de la robustesse de notre approche face à cette difficulté supplémentaire. Pour plus de détails, la matrice de confusion est fournie dans la Figure 5.7.

	HG1	HG2	HG3	HG4	HG5	HG6	HG7	HG8	HG9	HG10	HG11	HG12	HG13
HG1	92.9	7.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HG2	6.7	80.0	0.0	6.7	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HG3	0.0	0.0	92.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.1
HG4	0.0	6.7	0.0	86.7	0.0	0.0	0.0	0.0	6.7	0.0	0.0	0.0	0.0
HG5	0.0	6.7	0.0	0.0	66.7	0.0	0.0	0.0	0.0	0.0	0.0	20.0	6.7
HG6	0.0	0.0	0.0	0.0	0.0	85.7	0.0	0.0	0.0	0.0	7.1	0.0	7.1
HG7	0.0	0.0	6.7	0.0	0.0	0.0	93.3	0.0	0.0	0.0	0.0	0.0	0.0
HG8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
HG9	0.0	0.0	0.0	0.0	0.0	0.0	8.3	0.0	83.3	0.0	0.0	0.0	8.3
HG10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	92.9	0.0	0.0	7.1
HG11	0.0	6.7	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	86.7	0.0	0.0
HG12	0.0	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.7	0.0	86.7	0.0
HG13	0.0	0.0	6.7	0.0	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	86.7

FIGURE 5.7 – Notre matrice de confusion sur l'ensemble de données LMDHG.

5.3 Interaction dans un environnement 3D

Dans cette section, nous présentons une application dans le domaine de l'interaction Homme-Machine. Il s'agit d'une application réalisée dans le cadre d'un module dispensé à des étudiants de $5^{ème}$ année de l'INSA de Rennes. Il s'agit de créer un système de détection de commandes gestuelles, basé sur le dispositif de capture Leap Motion. L'objectif à la fin de ce module est de proposer un jeu développé sous Unity où l'utilisateur contrôle un personnage virtuel avec des commandes gestuelles.

Comme illustré dans la Figure 5.8, le jeu consiste à faire éviter au personnage de heurter les différents obstacles sur la scène. Si aucune action n'est spécifiée, le personnage par défaut effectue une marche en avant. Dès que le système de reconnaissance détecte et reconnaît une commande de l'utilisateur, le personnage réalise l'action associée. En configuration de base, ce personnage peut se mouvoir à gauche ou à droite, effectuer un saut ou une roulade, reprendre la marche en avant. Ceci constitue donc cinq classes de gestes à modéliser et à reconnaître.

FIGURE 5.8 – Illustration de l'interface de jeu.

Le flot à analyser étant non segmenté, le système de détection utilisé est basé sur l'approche **CuDi3D** développée dans le cadre de la OAD (cf. chapitre 4). De plus, s'agissant d'une détection des gestes dynamiques de la main et non pas des actions corps-complet, nous avons modifié l'approche **CuDi3D** pour y intégrer la représentation introduite dans la section 5.2.1. Ainsi, nous avons défini cinq classifieurs, correspondant aux tailles curvilignes des cinq classes à détecter, et nous avons adapté la procédure de calcul des tailles curvilignes de chacun.

Sur l'interface illustrée dans la Figure 5.8 (en haut à gauche), nous affichons les actions

potentielles détectées en temps réel ainsi que la distance curviligne cumulée jusque-là. En haut à droite de cette illustration, nous affichons la classe effectivement détectée. Avec cette affichage nous offrons aux étudiants la possibilité de visualiser le résultat des choix et des modifications qu'ils ont à faire en personnalisant ce jeu. En effet, nous avons offert plusieurs leviers sur lesquels il est possible d'agir pour améliorer les performances de détection.

Un premier levier est le choix des articulations à considérer lors de la modélisation du geste. Comme présenté dans la section 5.2.1, il est possible de considérer une ou deux mains et choisir ainsi que les articulations à utiliser. Ceci permet aux étudiants d'évaluer le rôle des articulations retenues en fonction de l'ensemble de gestes choisis et d'en mesurer l'impact sur la performance de leur système. Un deuxième levier est le choix des classes de gestes à détecter. Ceci permet aux étudiants d'une part, d'avoir des retours utilisateur concernant la difficulté ou non d'effectuer un geste donné et d'autre part, de voir si la variabilité inter-classes est suffisante pour que le classifieur ne se trompe pas de détections. Un troisième levier est le choix de la stratégie de combinaison des décisions locales émises par chaque classifieur (cf. section 4.2.2.3). En effet, ceci offre la possibilité aux étudiants de comparer l'efficacité de leurs techniques de combinaisons des décisions locales avec celle proposée par notre approche.

5.4 Animation temps réel d'avatars

Il s'agit d'une autre application qui a été menée au cours d'un stage effectué à Mime-TIC par Alexandre Bonneau, étudiant en $3^{ème}$ année de Licence en informatique à l'École Normale Supérieure de Rennes. L'objectif de ce stage est d'améliorer l'animation en temps réel d'un avatar à partir de mesures Kinect. En fait, on s'est mis dans la situation où un utilisateur anime un avatar en effectuant des actions corps-complet devant une caméra Kinect. Or, comme ces actions sont bruitées, l'idée est d'afficher, non pas les actions effectuées par l'utilisateur, mais plutôt les mouvements d'une base de gestes pré-enregistrés, sans artefact et parfaitement adaptés à l'humain virtuel.

Le choix de l'animation à rejouer est fait sur la base des actions détectées par un système utilisant l'approche **E-CuDi3D** de la OAD présentée dans le chapitre 4. Pour assurer un compromis entre la précocité et la qualité du rendu visuel, il fallait trouver un moyen d'afficher correctement les postures prises par le sujet sur la base des décisions émises par le système. Ainsi, il a été question de combiner les décisions intermédiaires émises par chacun des modèles composant le système de détection précoce. Ci-après, nous décrivons plus en détail la problématique considérée. Nous présentons par la suite la piste de recherche explorée, à savoir la combinaison des décisions locales, ainsi que les résultats

préliminaires obtenus.

5.4.1 Problématique

Pour animer un personnage, il est possible de définir manuellement les mouvements qu'il va effectuer grâce à des logiciels de création, ou bien en effectuant une capture des mouvements. Cette dernière méthode donne des résultats très réalistes puisqu'elle affiche le mouvement capturé sur un sujet réel. Elle est néanmoins dépendante du système d'acquisition. En particulier, alors que la Kinect est un outil de capture peu onéreux et facile d'utilisation, elle présente l'inconvénient de fournir des données bruitées, générant souvent un mouvement approximatif du mouvement réalisé.

Ainsi, la problématique considérée est l'amélioration en temps réel de l'affichage (animation) des personnages (avatars) sur la base des données fournies par la Kinect. La piste de recherche retenue est de baser cet affichage sur des actions pré-enregistrées en les exécutant à partir des détections d'actions issues du système basé sur l'approche de détection précoce **E-CuDi3D** (cf. section 4.3.2). Autrement dit, l'objectif est de rejouer une action pré-enregistrée, parfaitement débruitée, aussitôt que le système de détection aurait identifié l'action en cours.

En outre, comme présenté dans la section 4.3.2, le système de détection émet des décisions locales au fur et à mesure que de nouvelles frames sont disponibles avant d'émettre la décision finale. Ces décisions locales peuvent d'une part, ne pas correspondre à la classe finale prédite (dans le cas où plusieurs classes partagent des débuts communs) et d'autre part, être différentes pour les modèles qui composent le système de détection. Ainsi, pour augmenter la réactivité nous nous basons sur les décisions locales de chaque modèle en combinant les mouvements des classes potentiellement détectées à chaque frame. Cette solution est détaillée dans la section suivante.

5.4.2 Approche de combinaison des décisions

Cette approche consiste à utiliser les actions potentielles, fournies en temps réel par le système de détection, pour afficher, à chaque instant, une posture proche de celle de l'utilisateur. En particulier, à chaque instant, les postures correspondant aux actions potentielles sont mélangées (*blending*) de manière à obtenir une posture finale proche de la posture réelle.

Lors de ce mélange, chaque posture est pondérée par le score de confiance que donne le modèle local l'ayant prédite. Ceci permet de se rapprocher au mieux de la trajectoire que fait l'utilisateur même si les classes d'actions prédites sont différentes de celle que l'utilisateur veut faire. Par exemple, si parmi les classes à détecter, il y a un service de

tennis, un service de volley et le salut à un ami et que l'utilisateur lève la main gauche (mouvement commun à ces trois actions), alors ces trois mouvements sont détectés lors des premières frames. Les mélanger et les afficher sur cette période de temps ferait lever la main gauche de l'avatar, ce qui est souhaité.

Pour ce qui est de la technique de combinaison, il est possible de synchroniser les mouvements candidats lors du mélange en se basant sur le concept de distance curviligne. En effet, vu que chaque modèle local fournit aussi la distance curviligne cumulée à la frame courante, il est possible de situer la posture dans chacune des séquences représentatives de l'action prédite. Ainsi, les distances curvilignes cumulées sont calculées pour chaque séquence représentative, en faisant correspondre chaque valeur avec une posture de la classe d'action associée. Le processus global est schématisé dans la Figure 5.9.

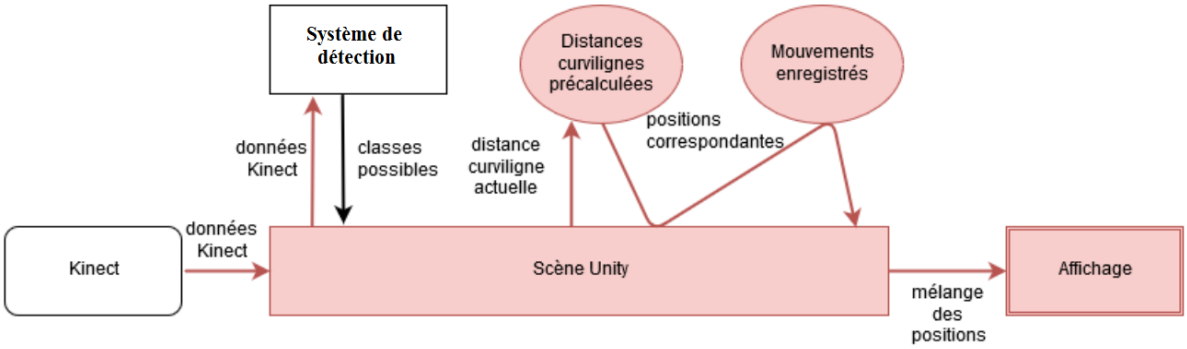

FIGURE 5.9 – Schématisation du processus global.

Le processus illustré dans la Figure 5.9 commence par la récupération des données de la Kinect (positions de toutes les articulations). Ces données sont ensuite envoyées au système de détection qui extrait des représentations suivant différentes fenêtres curvilignes et renvoie à la scène Unity les classes prédites, les scores de confiance et les distances curvilignes que fournit chaque modèle. Sur la base de ces informations, les postures sont choisies parmi les mouvements candidats en tentant de rapprocher la distance curviligne ainsi obtenue et celle cumulée jusque-là par le système. Ces positions sont ensuite combinées et la posture finale est affichée.

5.4.3 Résultats préliminaires et discussion

Pour l'évaluation de l'approche d'animation proposée, nous avons utilisé les mêmes actions présentes dans la base de données MSRC-12 [FMKN12] (cf. chapitre 4) qui sont au nombre de 12 classes. Néanmoins, cette base n'est composée que des coordonnées cartésiennes 3D de 20 articulations, sans les angles inter-segmentaires. Ceci complexifie notamment le mélange mais aussi l'animation sous Unity vu que ces données angulaires

sont primordiales. De ce fait, une recapture complète d'une séquence représentative de chaque classe d'action a été effectuée via la Kinect.

Nous avons alors conçu deux prototypes d'animation, correspondant à deux expérimentations. Dans chacun des deux cas, nous avons créé deux avatars dans la scène, l'un animé avec les données brutes fournies par le système de capture (Kinect) et l'autre qui rejoue les actions pré-enregistrées lorsque le système de détection identifie une action donnée. La différence entre ces deux prototypes réside dans le système de détection utilisé. En effet, la première expérimentation consiste à baser le système d'animation sur les résultats rendus par la méthode de détection **CuDi3D** dans une configuration de détection non-précoce. Au contraire, lors de la deuxième expérimentation, le second avatar est animé avec le résultat du mélange des décisions précoces de l'approche **E-CuDi3D** comme expliqué précédemment à la section 5.4.2. Nous illustrons les résultats d'animation obtenus pour la classe d'action "s'incliner" lors de la première et deuxième expérimentation dans respectivement la Figure 5.10 et la Figure 5.11.

Globalement, sur les Figures 5.10 et 5.11, il est possible de noter que l'action pré-enregistrée donne un meilleur effet d'un point de vue animation (avatar de gauche) que celui donné en rejouant les données brutes (avatar de droite). En effet, lors de l'exécution du geste s'incliner, le bras de l'avatar de droite se tord complètement ainsi que sa jambe,

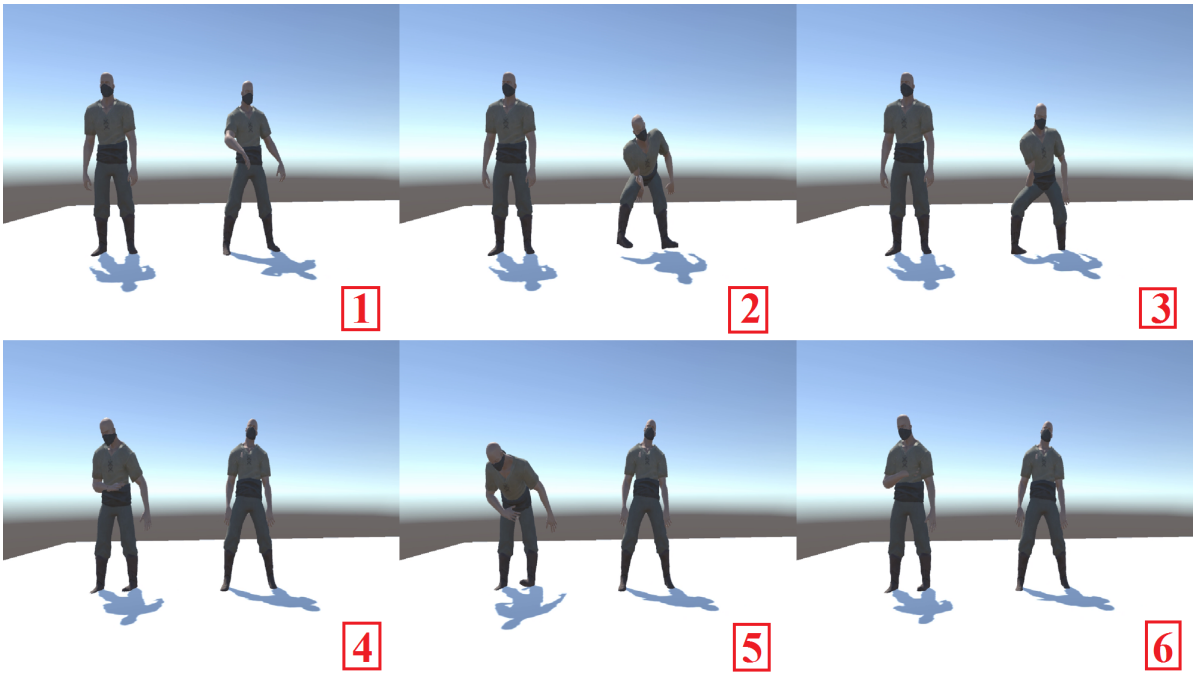

FIGURE 5.10 – Résultats d'animation d'un avatar (à gauche) pour l'action "s'incliner" au moyen de la méthode **CuDi3D** en détection non-précoce. L'avatar de droite correspond au rejeu des données brutes.

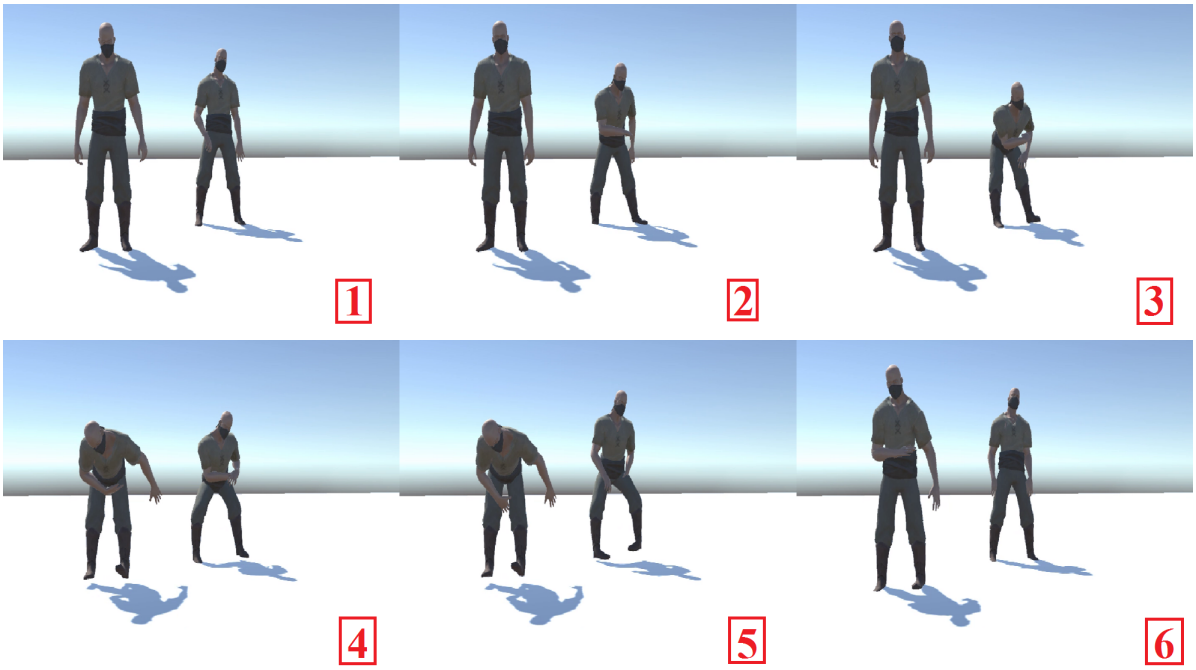

FIGURE 5.11 – Résultats d'animation d'un avatar (à gauche) pour l'action "s'incliner" au moyen du mélange des décisions précoces. L'avatar de droite correspond au rejeu des données brutes.

donnant au final une animation de qualité médiocre. A l'opposé, l'avatar de gauche effectue cette action de façon propre et réaliste, sans provoquer de perturbation au niveau de l'avatar. Cette première constatation confirme le fait qu'animer un avatar en rejouant directement les données brutes n'est pas envisageable au vu de la grande perte en termes de qualité d'animation.

De plus, il est possible de relever dans la Figure 5.10 qu'il existe une latence significative (de l'ordre d'une seconde) entre le moment où l'utilisateur effectue son action (image 1) et celui où l'avatar de gauche rejoue l'action pré-enregistrée correspondante (image 4). En effet, l'approche **CuDi3D**, utilisée dans cette première expérimentation, fournie un résultat bien après le début de l'action, voire à la fin de cette action. Cette latence pénalise le sentiment d'immersion, très important dans ce genre d'application.

Au contraire, dans la Figure 5.11, correspondant à la deuxième expérimentation, il est possible de voir que l'avatar de gauche rejoue l'action détectée par le classifieur, en commençant à la frame correspondant à la distance curviligne cumulée. Ceci permet de rattraper le retard dû à la détection et résulte sur un quasi alignement des deux performances. Ainsi, pour avoir un rendu réaliste, il est possible d'animer l'avatar avec les données brutes tant que le classifieur n'a pas commencé à donner les premières décisions locales. Ceci éviterait notamment d'avoir un avatar qui passe d'une position repos à la

moitié du geste comme c'est visible sur les images 3 et 4 de la Figure 5.11. Une fois que les premières décisions sont émises, il faudrait combiner la trajectoire brute et celle pré-enregistrée de manière à passer progressivement à un rejeu total de l'action pré-enregistrée.

Dans les Figure 5.12 et Figure 5.13, nous donnons plus d'exemples d'animation. La Figure 5.12 contient les frames obtenues lors de la première expérimentation pour trois classes d'actions, à savoir un coup de pied (G12), mettre des lunettes (G4) et translater le bras horizontalement (G3). La Figure 5.13 contient les frames obtenues lors de la seconde expérimentation pour trois autres classes d'actions, à savoir jeter un objet (G8), soulever les bras en haut (G5) et translater le bras horizontalement (G3).

Ces illustrations additionnelles permettent de mettre en évidence deux points. D'une part, en comparant les frames des deux figures, il est possible de relever le gain en terme de latence que permet de réaliser l'utilisation d'un mélange des décisions locales au lieu d'utiliser l'approche **CuDi3D** telle quelle dans une configuration de détection non-précoce. D'autre part, en analysant les frames représentées dans la Figure 5.13, notamment la frame 4 de l'action G5 (soulever les bras en haut), il est possible de noter qu'un mélange a bien lieu entre les trajectoires de différentes actions. Ce mélange permet en effet de réduire davantage la latence tout en produisant des rendus réalistes. Il est ainsi possible de conclure que ces résultats préliminaires attestent de l'intérêt de cette approche d'animation. Des travaux vont d'ailleurs être menés dans ce sens à la suite de ces expérimentations en conjonction avec les deux équipes.

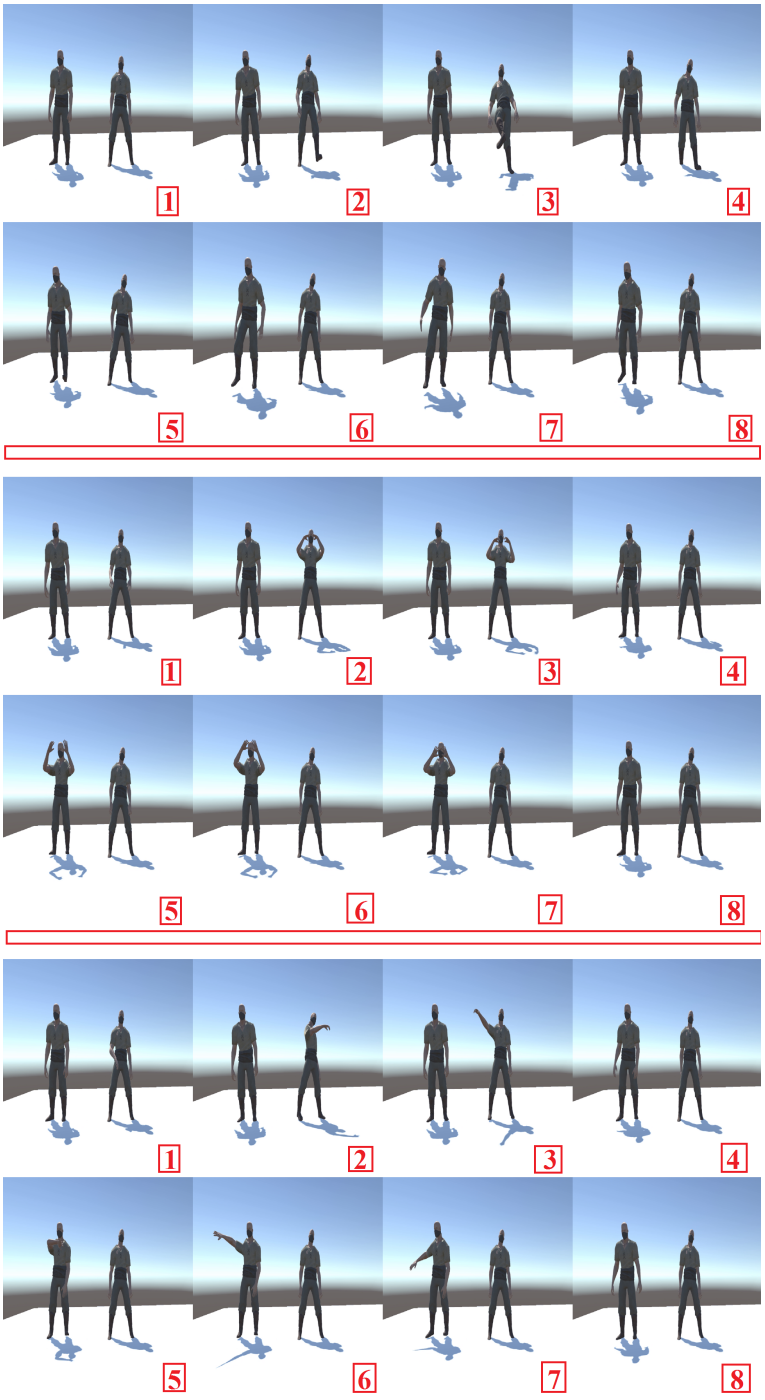

FIGURE 5.12 – Illustration de frames de la première expérimentation pour trois classes d'actions : un coup de pied (G12), mettre des lunettes (G4) et translater le bras horizontalement (G3), de haut en bas et de gauche à droite. Chaque action est représentée en huit frames réparties sur deux lignes.

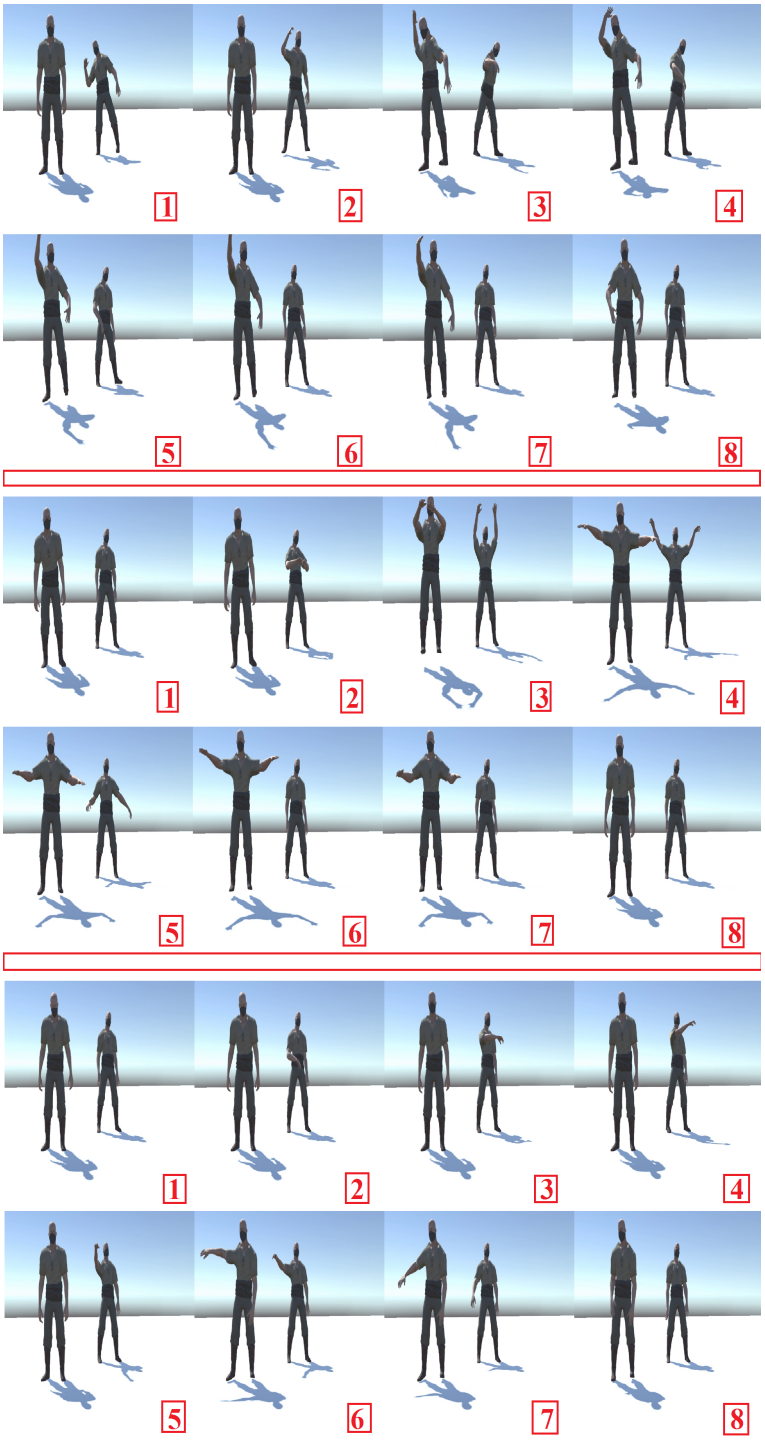

FIGURE 5.13 – Illustration de frames de la seconde expérimentation pour trois classes d'actions : jeter un objet (G8), soulever les bras en haut (G5) et translater le bras horizontalement (G3), de haut en bas et de gauche à droite. Chaque action est représentée en huit frames réparties sur deux lignes.

5.5 Conclusion

Dans ce dernier chapitre, nous avons présenté trois travaux dans lesquels nous avons appliqué les approches de reconnaissance et de détection proposées dans les chapitres précédents.

D'une part, nous avons proposé d'adresser la problématique de reconnaissance des gestes dynamiques de la main, une problématique connexe à celle de reconnaissance d'actions corps-complet. En particulier, nous avons réussi à faire un parallèle entre les doigts et les trajectoires des articulations du corps-complet afin d'exploiter la puissance de modélisation des descripteurs conçus précédemment pour les actions corps-complet. Ce faisant, nous voulons aller au bout de la fusion des techniques permettant la modélisation de trajectoires créées par l'humain lors de la performance d'un geste qu'il soit 2D, 3D corps-complet ou 3D effectué avec la main. D'autre part, nous avons aussi montré que les approches de reconnaissance et de détection d'actions que nous avons conçues ont des applications prometteuses dans le domaine d'animation d'avatar.

Chapitre 6

Conclusion & Perspectives

6.1 Conclusion

Le développement d'approches à même de modéliser, reconnaître et détecter les actions 3D à base de données squelettiques, permet d'imaginer aujourd'hui de nouvelles modalités d'interaction Homme-Machine. Dans ce contexte, les travaux menés dans cette thèse sont structurés autour de deux éléments principaux. D'un côté, nous avons proposé d'aborder la problématique de reconnaissance d'actions 3D en tirant profit d'avancées réalisées dans d'autres domaines de la reconnaissance de formes, notamment celui de la reconnaissance de tracés manuscrits 2D. D'un autre côté, nous avons souhaité inscrire ces travaux dans une démarche qui peut être caractérisée comme explicite, revendiquant une certaine "transparence" par opposition à beaucoup d'approches de l'état de l'art qui s'appuient sur une démarche de type "boîte noire".

L'objectif de nos travaux de recherche est donc de concevoir une approche "transparente" originale apte à détecter en temps réel l'occurrence d'une action, dans un flot non segmenté et idéalement le plus tôt possible. Le caractère explicite (transparent) des stratégies mises en place dans l'approche nous a permis de l'affiner et de l'optimiser progressivement pour aboutir à une solution robuste et générique. Pour parvenir à la mise en place d'une telle approche, nous avons adressé plusieurs sous-problèmes intermédiaires. Pour chacun de ces sous-problèmes, nous avons explicité les choix des stratégies retenues : (1) adresser méthodiquement les sous-problématiques permettant d'atteindre l'objectif final énoncé plus haut, (2) identifier explicitement les difficultés majeures à relever pour chaque sous-problématique et (3) proposer des solutions ciblant les difficultés relevées en se basant notamment sur le savoir-faire existant dans la communauté de la reconnaissance des formes 2D.

Nous avons d'abord présenté les travaux de l'état de l'art relatifs à la reconnais-

sance et la détection d'actions squelettiques selon quatre aspects distincts du processus de reconnaissance. Le premier aspect porte sur le type de données utilisées en entrée : les coordonnées cartésiennes (absolues ou relatives), les coordonnées angulaires (absolues ou relatives), les relations géométriques ou la combinaison de plusieurs modalités. Le deuxième aspect est relatif à la manière d'appréhender la nature séquentielle des données gestuelles. Le troisième aspect concerne la quantité de données nécessaire pour entraîner le modèle en question. Le dernier aspect est lié à la distinction entre les approches de reconnaissance d'actions dans un flot pré-segmenté et les approches de détection permettant la reconnaissance d'actions dans un flot continu (non segmenté).

Nous avons ensuite fait émerger trois problématiques majeures liées aux actions 3D. La première problématique consiste à réfléchir à une nouvelle approche pour modéliser et reconnaître **une action pré-segmentée**. Pour cette problématique, nous avons identifié trois difficultés majeures : la variabilité morphologique, les corrélations spatiales et le séquencement temporel. La deuxième problématique consiste à développer une approche permettant de reconnaître **une action dans un flot non segmenté** : sans connaissance a priori du début ni de la fin de l'action. Cette deuxième problématique, dite **détection en-ligne**, est plus complexe que la première dans la mesure où il s'agit de reconnaître des actions tout en les segmentant dans un flot continu de mouvements, qui peut être décomposé (ou non) par des positions de repos potentiellement identifiables. En plus des trois difficultés de reconnaissance d'actions pré-segmentées, il est nécessaire, pour cette deuxième problématique, d'adresser trois autres difficultés : la variabilité temporelle, la variabilité spatiale inter-classes et la variabilité spatiale intra-classe. La troisième et dernière problématique consiste à concevoir une approche permettant la **caractérisation précoce d'une action avec très peu d'informations**, c'est-à-dire le début de cette action. Pour cette problématique, une contrainte supplémentaire est à considérer : il faut être capable de décider à partir de quand le système peut émettre une hypothèse de reconnaissance au plus tôt sans se tromper.

Pour résoudre la première problématique portant sur la **reconnaissance d'actions pré-segmentées**, nous avons proposé, dans un premier temps, de concevoir une représentation permettant de modéliser une action 3D pré-segmentée à base de données squelettiques. Comme présenté dans le chapitre 3, nous avons exploré deux pistes différentes pour mener le transfert du savoir-faire 2D vers la 3D, chacune résultant sur une représentation différente **des trajectoires de mouvements** déduites des données squelettiques. Une première manière d'opérer ce transfert consiste à ramener le problème de modélisation d'actions 3D dans un espace de représentation 2D. Ceci permet, notamment, d'appliquer directement les techniques de modélisation de tracés manuscrits 2D. La deuxième manière de transférer le potentiel de modélisation des techniques 2D aux patterns 3D consiste à

transposer les descripteurs 2D dans l'espace 3D. Nous avons en effet conçu un nouveau jeu de descripteurs, dénommé **HIF3D**, pour représenter des actions 3D en s'inspirant de descripteurs 2D. Les expérimentations conduites sur trois bases de données d'actions pré-segmentées ont permis de montrer la pertinence des approches proposées et par voie de conséquence l'intérêt du transfert mené.

Dans un second temps, pour répondre à la problématique de **détection en-ligne d'actions squelettiques dans un flot non segmenté**, nous avons proposé une approche à trois étages, dénommée **CuDi3D**, permettant d'adresser les trois difficultés identifiées. Nous avons notamment introduit le concept de **segmentation curviligne des trajectoires** en opposition au concept existant de fenêtres temporelles. Ce nouveau concept de segmentation, qui permet notamment de faire face à la variabilité temporelle, a la particularité de considérer les frames en entrée, non pas comme un flot temporel, mais comme un flot indexé sur la quantité de mouvements. Par ailleurs, nous avons conçu un système de décision innovant, combinant les détections locales de plusieurs classifieurs spécialisés lancés en parallèle, permettant d'adresser efficacement les variabilités spatiales inter-classes et les variabilités spatiales intra-classe. La validation expérimentale exhaustive, que nous avons menée sur trois bases de données de l'état de l'art, a permis de mettre en évidence l'intérêt et l'apport de notre approche.

Dans un troisième temps, nous avons considéré le problème de **détection précoce d'actions 3D dans un flot non segmenté**. Au cours de cette tâche, le plus important est de réussir à identifier une classe au plus tôt. L'approche proposée, dénommée **E-CuDi3D**, se base sur trois modèles concurrents, où chacun est en fait une variante de l'approche de la OAD **CuDi3D**. Par ailleurs, nous avons aussi proposé une extension de l'approche de la OAD **CuDi3D** pour adresser plus efficacement la problématique de reconnaissance d'actions 3D pré-segmentées. Des résultats prometteurs sont rapportés avec cette approche.

Dans un dernier chapitre, nous avons montré le degré de généricité de notre approche pour modéliser des gestes de différentes natures et dans différents contextes. En particulier, nous avons évalué les performances du jeu de descripteurs **HIF3D** pour aborder la problématique de reconnaissance des gestes dynamiques de la main. Nous avons aussi présenté une application réalisée dans le cadre d'un projet proposé aux étudiants de $5^{ème}$ année informatique de l'INSA de Rennes, dont l'objectif est de déplacer un humain virtuel dans un environnement 3D via le Leap Motion. Enfin, nous avons présenté les résultats préliminaires d'une étude perceptive menée au cours d'un stage effectué à MimeTIC par un étudiant en $3^{ème}$ année de Licence. L'objectif de cette étude est de mesurer l'intérêt, par rapport à l'utilisation des données brutes, de rejouer des mouvements pré-enregistrés pour animer un avatar. Le choix du mouvement à rejouer s'effectue sur la base des ré-

sultats de détection rendus par l'approche **E-CuDi3D**. Les résultats obtenus sont très prometteurs et permettent d'envisager d'explorer davantage cette piste.

6.2 Perspectives

A travers les résultats présentés dans ce manuscrit, nous avons montré l'intérêt d'aborder la problématique de reconnaissance et de détection d'actions 3D à travers le prisme des techniques de modélisation 2D. Néanmoins, cette piste de recherche n'est pas encore totalement épuisée. En effet, plusieurs améliorations peuvent être apportées. Ainsi, nous présentons ici quelques perspectives à notre travail :

— Une première amélioration est le renforcement de la généricité de la représentation conçue à base des descripteurs **HIF3D** au moyen d'une adaptation automatique de cette représentation à l'application ciblée. Nous avons notamment montré au cours de cette thèse que le jeu de descripteurs **HIF3D** est performant pour modéliser des trajectoires de différentes natures (des trajectoires issues des actions corps-complet, des gestes dynamiques de la main, des trajectoires issues des actions haut du corps). L'idée est de développer un module, en amont du processus d'extraction des descripteurs, qui ne retiendrait, pour l'application ciblée, que les trajectoires des articulations les plus discriminantes pour les classes de gestes considérées. Ceci permettrait donc à notre représentation d'être indépendante par rapport aux systèmes de capture utilisés (Leap motion, Kinect, etc.) et par rapport aux gestes à reconnaître. De plus, comme évoqué dans la chapitre 3, une sélection des descripteurs peut aussi améliorer les performances d'une représentation. Il serait donc aussi intéressant d'inclure dans ce module, une sélection automatique des descripteurs en fonction de l'application ciblée, voire même pour chaque ensemble de classes de gestes considérées. Un tel module permettrait une plus forte généricité de notre représentation en puisant dans un ensemble réduit de descripteurs (**HIF3D**) ceux les plus à même de distinguer les gestes considérés.

— Une autre perspective intéressante porte sur la capacité à faire évoluer un système de reconnaissance en fonction des performances gestuelles réalisées par le sujet/participant. Par exemple, en employant un système d'interaction à base de commandes gestuelles, l'utilisateur aurait tendance à changer sa performance au fur et à mesure qu'il s'habitue au système. L'objectif de développer un système évolutif est de pouvoir tenir compte de ce changement et de mettre à jour de manière implicite le moteur de reconnaissance. Cette proposition est facilitée d'une part, par la transparence de notre représentation de gestes et, d'autre part, par la dimension réduite de cette représentation. La transparence de l'approche permet

de modifier simplement sa structure pour inclure des modules permettant la prise en compte de ces changements. La dimension réduite de la représentation permet d'envisager le ré-apprentissage fréquent du moteur de reconnaissance et d'assurer donc son évolutivité.

— Une dernière perspective à nos travaux de recherche concerne l'amélioration des techniques d'animation à base de détection d'actions. En, effet, dans l'approche présentée dans le chapitre 5, nous avons fusionné les animations sur la base des décisions locales des modèles curvilignes de détection. Une perspective est d'anticiper encore en commençant d'intégrer en plus les données brutes du sujet dans l'animation du personnage virtuel. Il serait aussi intéressant de mener une étude perceptive pour évaluer quantitativement si l'amélioration en termes de rendu visuel par rapport à un affichage classique est significative pour les utilisateurs.

Publications de l'auteur

Conférences internationales

Said Yacine Boulahia, Eric Anquetil, Franck Multon, Richard Kulpa. Dynamic hand gesture recognition based on 3D pattern assembled trajectories. *7th IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA 2017)*, pp. 1-6, November 2017, Montreal, Canada.

Said Yacine Boulahia, Eric Anquetil, Richard Kulpa, Franck Multon. 3D Multistroke Mapping (3DMM) : Transfer of hand-drawn pattern representation for skeleton-based gesture recognition. *12th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2017)*, pp. 462-467, May 2017, Washington D.C, USA.

Said Yacine Boulahia, Eric Anquetil, Richard Kulpa, Franck Multon. HIF3D : Handwriting-inspired features for 3D skeleton-based action recognition. *23rd IEEE International Conference on Pattern Recognition (ICPR 2016)*, pp. 985-990, Dec 2016, Cancun, Mexico.

Conférences nationales

Said Yacine Boulahia, Eric Anquetil, Franck Multon, Richard Kulpa. Détection précoce d'actions squelettiques 3D dans un flot non segmenté à base de modèles curvilignes. *Congrés Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP 2018)*, June 2018, Paris, France.

Bibliographie

[AA11] Abdullah Almaksour and Eric Anquetil. Improving premise structure in evolving takagi–sugeno neuro-fuzzy classifiers. *Evolving Systems*, 2(1) :25–33, 2011.

[AL97] E Anquetil and Guy Lorette. Perceptual model of handwriting drawing. application to the handwriting segmentation problem. In *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, volume 1, pages 112–117. IEEE, 1997.

[BAM13] Victoria Bloom, Vasileios Argyriou, and Dimitrios Makris. Dynamic feature selection for online action recognition. In *Proceedings of the 4th International Workshop on Human Behavior Understanding, 2013*, pages 64–76, 2013.

[BAM16] Victoria Bloom, Vasileios Argyriou, and Dimitrios Makris. Hierarchical transfer learning for online recognition of compound actions. *Computer Vision and Image Understanding*, 144 :62–72, 2016.

[BAM17] Victoria Bloom, Vasileios Argyriou, and Dimitrios Makris. Linear latent low dimensional space for online early action recognition and prediction. *Pattern Recognition*, 72 :532–547, 2017.

[BDH96] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4) :469–483, 1996.

[BK59] Richard Bellman and Robert Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2) :1–9, 1959.

[BKK17] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Real-time online action detection forests using spatio-temporal contexts. In *Proceedings of*

the IEEE Winter Conference on Applications of Computer Vision, 2017, pages 158–167, 2017.

[BLMC09] Oliver Brdiczka, Matthieu Langet, Jérôme Maisonnasse, and James L Crowley. Detecting human behavior models from multimodal observation in a smart home. *IEEE Transactions on Automation Science and Engineering*, 6(4) :588–597, 2009.

[BMA12] Victoria Bloom, Dimitrios Makris, and Vasileios Argyriou. G3d : A gaming action dataset and real time action recognition evaluation framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2012*, pages 7–12, 2012.

[BMA14] Victoria Bloom, Dimitrios Makris, and Vasileios Argyriou. Clustered spatio-temporal manifolds for online action recognition. In *Proceedings of the 22nd IEEE International Conference on Pattern Recognition, 2014*, pages 3963–3968, 2014.

[BPSW70] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1) :164–171, 1970.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1) :5–32, 2001.

[Bre18] *Brekel Affordable Motion Capture Tools*, Available at `http://brekel.com/kinect-2-details-specifications-observations/`. 2018 (accessed April 23, 2018).

[BS15] Somar Boubou and Einoshin Suzuki. Classifying actions based on histogram of oriented velocity vectors. *Journal of Intelligent Information Systems*, 44(1) :49–65, 2015.

[CB95] Lee W Campbell and Aaron F Bobick. Recognition of human body motion using phase space constraints. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 624–630. IEEE, 1995.

[CBB+13] Yu Cao, Daniel Barrett, Andrei Barbu, Siddharth Narayanaswamy, Haonan Yu, Aaron Michaux, Yuewei Lin, Sven Dickinson, Jeffrey Mark Siskind, and Song Wang. Recognize human activities from partially observed videos. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2658–2665. IEEE, 2013.

[CC14] Kyunghyun Cho and Xi Chen. Classifying and visualizing motion capture sequences using deep neural networks. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 2, pages 122–130. IEEE, 2014.

[CE11] James Charles and Mark Everingham. Learning shape models for monocular human pose estimation from the microsoft xbox kinect. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1202–1208. IEEE, 2011.

[CJK16] Chen Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. A real-time human action recognition system using depth and inertial sensor fusion. *IEEE Sensors Journal*, 16(3) :773–781, 2016.

[CK13] Xi Chen and Markus Koskela. Online rgb-d gesture recognition with extreme learning machines. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 467–474. ACM, 2013.

[CL11a] Chih-Chung Chang and Chih-Jen Lin. LIBSVM : A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2 :27 :1–27 :27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[CL11b] Chih-Chung Chang and Chih-Jen Lin. Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3) :27, 2011.

[cnb18] *Dynamic Time Warping*, Available at `http://www.cnblogs.com/luxiaoxun/archive/2013/05/09/3069036.html`. 2018 (accessed April 23, 2018).

[COK+13] Rizwan Chaudhry, Ferda Ofli, Gregorij Kurillo, Ruzena Bajcsy, and René Vidal. Bio-inspired dynamic 3d discriminative skeletal features for human action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 471–478, 2013.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3) :273–297, 1995.

[CW00] Claude Cadoz and Marcelo M Wanderley. Gesture-music, 2000.

[DA13] Adrien Delaye and Eric Anquetil. Hbf49 feature set : A first unified baseline for online symbol recognition. *Pattern Recognition*, 46(1) :117–130, 2013.

[DBP+17] Maxime Devanne, Stefano Berretti, Pietro Pala, Hazem Wannous, Mohamed Daoudi, and Alberto Del Bimbo. Motion segment decomposition of rgb-d sequences for human behavior understanding. *Pattern Recognition*, 61 :222–233, 2017.

[DKOH15] Emel Demircan, Dana Kulic, Denny Oetomo, and Mitsuhiro Hayashibe. Human movement understanding. *IEEE Robotics & Automation Magazine*, 22(3) :22–24, 2015.

[DLCJ16] Wenwen Ding, Kai Liu, Fei Cheng, and J. Zhang. Learning hierarchical spatio-temporal pattern for human activity prediction. *Journal of Visual Communication and Image Representation*, 35 :103–111, 2016.

[DLCZ15] Wenwen Ding, Kai Liu, Fei Cheng, and Jin Zhang. Stfc : Spatio-temporal feature chain for skeleton-based human action recognition. *Journal of Visual Communication and Image Representation*, 26 :329–337, 2015.

[DM15] Marc Dupont and Pierre-François Marteau. Coarse-dtw for sparse time series alignment. In *International Workshop on Advanced Analytics and Learning on Temporal Data*, pages 157–172. Springer, 2015.

[DSWV16] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeborre. Skeleton-based dynamic hand gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–9, 2016.

[DSWV+17] Quentin De Smedt, Hazem Wannous, Jean-Philippe Vandeborre, Joris Guerry, Bertrand Le Saux, and David Filliat. Shrec'17 track : 3d hand gesture recognition using a depth and skeletal dataset. In *10th Eurographics Workshop on 3D Object Retrieval*, 2017.

[DT06] James W Davis and Ambrish Tyagi. Minimal-latency human action recognition using reliable-inference. *Image and Vision Computing*, 24(5) :455–472, 2006.

[Dup17] Marc Dupont. *Glove-based gesture recognition for real-time outdoors robot control*. PhD thesis, Université de Bretagne Sud, 2017.

[DWB+15] Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE transactions on cybernetics*, 45(7) :1340–1352, 2015.

[DWP+15] Maxime Devanne, Hazem Wannous, Pietro Pala, Stefano Berretti, Mohamed Daoudi, and Alberto Del Bimbo. Combined shape analysis of human poses and motion units for action segmentation and recognition. In *Proceedings of the 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, 2015*, volume 7, pages 1–6, 2015.

[DWW15] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[EFV07] Alon Efrat, Quanfu Fan, and Suresh Venkatasubramanian. Curve matching, time warping, and light fields : New algorithms for computing similarity between curves. *Journal of Mathematical Imaging and Vision*, 27(3) :203–216, 2007.

[EMS16] Hugo Jair Escalante, Eduardo F Morales, and L Enrique Sucar. A naive bayes baseline for early gesture recognition. *Pattern Recognition Letters*, 73 :91–99, 2016.

[EMT+13] Chris Ellis, Syed Zain Masood, Marshall F Tappen, Joseph J LaViola, and Rahul Sukthankar. Exploring the trade-off between accuracy and observational latency in action recognition. *International Journal of Computer Vision*, 101(3) :420–436, 2013.

[ESH14] Georgios Evangelidis, Gurkirt Singh, and Radu Horaud. Skeletal quads : Human action recognition using joint quadruples. In *Proceedings of the IEEE International Conference on Pattern Recognition*, pages 4513–4518, 2014, 2014.

[FLO14] Tatsuya Fujii, Jae Hoon Lee, and Shingo Okamoto. Gesture recognition system for human-robot interaction and its application to robotic service task. In *Proc. of the International Multi-Conference of Engineers and Computer Scientists (IMECS)*, volume 1, 2014.

[FMKN12] Simon Fothergill, Helena Mentis, Pushmeet Kohli, and Sebastian Nowozin. Instructing people for training gestural interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1737–1746. ACM, 2012.

[Fuj00] Masahiro Fujita. Digital creatures for future entertainment robotics. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 801–806. IEEE, 2000.

[GBCC08] Scott A Green, Mark Billinghurst, XiaoQi Chen, and J Geoffrey Chase. Human-robot collaboration : A literature review and augmented reality approach in design. *International Journal of Advanced Robotic Systems*, 5(1) :1, 2008.

[GCR12] Weina Ge, Robert T Collins, and R Barry Ruback. Vision-based analysis of small groups in pedestrian crowds. *IEEE transactions on pattern analysis and machine intelligence*, 34(5) :1003–1016, 2012.

[GDOS12] Ye Gu, Ha Do, Yongsheng Ou, and Weihua Sheng. Human gesture recognition through a kinect sensor. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 1379–1384. IEEE, 2012.

[GFA06] Gutemberg Guerra-Filho and Yiannis Aloimonos. Understanding visuomotor primitives for motion synthesis and analysis. *Computer Animation and Virtual Worlds*, 17(3-4) :207–217, 2006.

[GMLW14] Ankur Gupta, Julieta Martinez, James J Little, and Robert J Woodham. 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2601–2608, 2014.

[GMM17] Benyamin Ghojogh, Hoda Mohammadzade, and Mozhgan Mokari. Fisherposes for human action recognition using kinect sensor data. *IEEE Sensors Journal*, 2017.

[GSK+11] Ross Girshick, Jamie Shotton, Pushmeet Kohli, Antonio Criminisi, and Andrew Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 415–422. IEEE, 2011.

[GTHES13] Mohammad Abdelaziz Gowayyed, Marwan Torki, Mohamed Elsayed Hussein, and Motaz El-Saban. Histogram of oriented displacements (hod) : Describing trajectories of human joints for action recognition. In *IJCAI*, pages 1351–1357, 2013.

[HOCB11] Brian Holt, Eng-Jon Ong, Helen Cooper, and Richard Bowden. Putting the pieces together : Connected poselets for human pose estimation. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1196–1201. IEEE, 2011.

[Hol93] Robert C Holte. Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1) :63–90, 1993.

[HRE+08] Chris Hecker, Bernd Raabe, Ryan W Enslow, John DeWeese, Jordan Maynard, and Kees van Prooijen. Real-time motion retargeting to highly varied user-created morphologies. *ACM Transactions on Graphics (TOG)*, 27(3) :27, 2008.

[HRHZ17] Fei Han, Brian Reily, William Hoff, and Hao Zhang. Space-time representation of people based on 3d skeletal data : A review. *Computer Vision and Image Understanding*, 158 :85–105, 2017.

[HTGES13] Mohamed E Hussein, Marwan Torki, Mohammad Abdelaziz Gowayyed, and Motaz El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*, volume 13, pages 2466–2472, 2013.

[HYWDLT14] Dong Huang, Shitong Yao, Yi Wang, and Fernando De La Torre. Sequential max-margin event detectors. In *European conference on computer vision*, pages 410–424. Springer, 2014.

[IBB] ESAT-PSI IBBT. Does human action recognition benefit from pose estimation ?

[IPS+12] Donato Impedovo, Giuseppe Pirlo, L Sarcinella, Erasmo Stasolla, and Claudia Adamita Trullo. Analysis of stability in static signatures using cosine similarity. In *Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on*, pages 231–235. IEEE, 2012.

[JCK13] Bongjin Jun, Inho Choi, and Daijin Kim. Local transform features and hybridization for accurate face and human detection. *IEEE transactions on pattern analysis and machine intelligence*, 35(6) :1423–1436, 2013.

[Joh73] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2) :201–211, 1973.

[JW14] Mithun George Jacob and Juan Pablo Wachs. Context-based hand gesture recognition for the operating room. *Pattern Recognition Letters*, 36 :196–203, 2014.

[JZPQ14] Xinbo Jiang, Fan Zhong, Qunsheng Peng, and Xueying Qin. Online robust action recognition based on a hierarchical model. *The Visual Computer*, 30(9) :1021–1033, 2014.

[KJ05] Richard Kulpa and Guy James. *Adaptation interactive et performante des mouvements d'humanoïdes synthétiques : aspects cinématique, cinétique et dynamique*. PhD thesis, Rennes, INSA, 2005.

[KZ07] Ana Kuzmanic and Vlasta Zanchi. Hand shape classification using dtw and lcss as similarity measures for vision-based gesture recognition system. In *EUROCON, 2007. The International Conference on" Computer as a Tool"*, pages 264–269. IEEE, 2007.

[LCS14] Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pages 689–704. Springer, 2014.

[LF12] Kang Li and Yun Fu. Arma-hmm : A new approach for early recognition of human activity. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1779–1782. IEEE, 2012.

[LKK16] Jonathan Feng-Shun Lin, Michelle Karg, and Dana Kulić. Movement primitive segmentation for human motion modeling : A framework for analysis. *IEEE Transactions on Human-Machine Systems*, 46(3) :325–339, 2016.

[LLX+16] Yanghao Li, Cuiling Lan, Junliang Xing, Wenjun Zeng, Chunfeng Yuan, and Jiaying Liu. Online human action detection using joint classification-regression recurrent neural networks. In *Proceedings of the European Conference on Computer Vision, 2016*, pages 203–220, 2016.

[LSX+17] Jun Liu, Amir Shahroudy, Dong Xu, Alex Kot Chichung, and Gang Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[LWQ13] Jiajia Luo, Wei Wang, and Hairong Qi. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1809–1816, 2013.

[LZXP17] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Skeleton-based action recognition with convolutional neural networks. In *Multimedia & Expo Workshops (ICMEW), 2017 IEEE International Conference on*, pages 597–600. IEEE, 2017.

[MAKD17] Marion Morel, Catherine Achard, Richard Kulpa, and Séverine Dubuisson. Automatic evaluation of sports motion : A generic computation of spatial and temporal errors. *Image and Vision Computing*, 64 :67–78, 2017.

[MAKD18] Marion Morel, Catherine Achard, Richard Kulpa, and Séverine Dubuisson. Time-series averaging using constrained dynamic time warping with tolerance. *Pattern Recognition*, 74 :77–89, 2018.

[MBR+09] Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stephane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th conference on autonomous robot systems and competitions*, volume 1, pages 59–65. IPCB : Instituto Politécnico de Castelo Branco, 2009.

[MHT15] Moustafa Meshry, Mohamed E Hussein, and Marwan Torki. Linear-time online action detection from 3d skeletal data using bags of gesturelets. *arXiv preprint arXiv :1502.01228*, 2015.

[MHT16] Moustafa Meshry, Mohamed E Hussein, and Marwan Torki. Linear-time online action detection from 3d skeletal data using bags of gesturelets. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.

[Mor17] Marion Morel. *Modélisation de séries temporelles multidimensionnelles. Application à l'évaluation générique et automatique du geste sportif.* PhD thesis, Université Pierre & Marie Curie, 2017.

[MR06] Meinard Müller and Tido Röder. Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of the*

2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, pages 137–146. Eurographics Association, 2006.

[MRC05] Meinard Müller, Tido Röder, and Michael Clausen. Efficient content-based retrieval of motion capture data. In *ACM Transactions on Graphics (ToG)*, volume 24, pages 677–685. ACM, 2005.

[MRC+07] Meinard Müller, Tido Röder, Michael Clausen, Bernhard Eberhardt, Björn Krüger, and Andreas Weber. Mocap database hdm05. *Institut für Informatik II, Universität Bonn*, 2 :7, 2007.

[MRR80] Cory Myers, Lawrence Rabiner, and Aaron Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(6) :623–635, 1980.

[MTI10] I Scott MacKenzie and Kumiko Tanaka-Ishii. *Text entry systems : Mobility, accessibility, universality*. Morgan Kaufmann, 2010.

[NS12] Sebastian Nowozin and Jamie Shotton. Action points : A representation for low-latency online human action recognition. *Microsoft Research Cambridge, Tech. Rep. MSR-TR-2012-68*, 2012.

[NSMG03] Michael Nielsen, Moritz Störring, Thomas B Moeslund, and Erik Granum. A procedure for developing intuitive and ergonomic gesture interfaces for hci. In *International gesture workshop*, pages 409–420. Springer, 2003.

[OBT13] Eshed Ohn-Bar and Mohan Trivedi. Joint angles similarities and hog2 for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 465–470, 2013.

[OCK+14] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy. Sequence of the most informative joints (smij) : A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation*, 25(1) :24–38, 2014.

[OL13] Omar Oreifej and Zicheng Liu. Hon4d : Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2013.

[OOH+05] Kei Okada, Takashi Ogura, Atsushi Haneda, Junya Fujimoto, Fabien Gravot, and Masayuki Inaba. Humanoid motion generation system on hrp2-jsk for daily life environment. In *Mechatronics and Automation, 2005 IEEE International Conference*, volume 4, pages 1772–1777. IEEE, 2005.

[Opt18] *OptiTrack*, Available at `http://www.optitrack.com/products/natnet-sdk`. 2018 (accessed April 23, 2018).

[PDLM15] Hossein Pazhoumand-Dar, Chiou-Peng Lam, and Martin Masek. Joint movement similarities for robust 3d action recognition using skeletal data. *Journal of Visual Communication and Image Representation*, 30 :10–21, 2015.

[PGKT10] Christian Plagemann, Varun Ganapathi, Daphne Koller, and Sebastian Thrun. Real-time identification and localization of body parts from depth images. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 3108–3113. IEEE, 2010.

[PL11] Chang-Beom Park and Seong-Whan Lee. Real-time 3d pointing gesture recognition for mobile robots with cascade hmm and particle filter. *Image and Vision Computing*, 29(1) :51–63, 2011.

[PLC16] Liliana Lo Presti and Marco La Cascia. 3d skeleton-based human action classification : A survey. *Pattern Recognition*, 53 :130–147, 2016.

[PLCSC14] Liliana Lo Presti, Marco La Cascia, Stan Sclaroff, and Octavia Camps. Gesture modeling by hanklet-based hidden markov model. In *Proceedings of the Asian Conference on Computer Vision*, pages 529–546, 2014, 2014.

[PNW12] Orasa Patsadu, Chakarida Nukoolkit, and Bunthit Watanapa. Human gesture recognition using kinect camera. In *Computer Science and Software Engineering (JCSSE), 2012 International Joint Conference on*, pages 28–32. IEEE, 2012.

[Ram91] Christophe Ramstein. *Analyse, représentation et traitement du geste instrumental : application aux instruments à clavier*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 1991.

[RMHM14] Hossein Rahmani, Arif Mahmood, Du Q Huynh, and Ajmal Mian. Real time action recognition using histograms of depth gradients and random decision forests. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 626–633. IEEE, 2014.

[Ryo11] Michael S Ryoo. Human activity prediction : Early recognition of ongoing activities from streaming videos. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1036–1043. IEEE, 2011.

[SC78] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1) :43–49, 1978.

[SDD12] Yale Song, David Demirdjian, and Randall Davis. Continuous body and hand gesture recognition for natural human-computer interaction. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(1) :5, 2012.

[SGW+13] Yang Song, Yu Gu, Peisen Wang, Yuanning Liu, and Ao Li. A kinect based gesture recognition algorithm using gmm and hmm. In *Biomedical Engineering and Informatics (BMEI), 2013 6th International Conference on*, pages 750–754. IEEE, 2013.

[SH80] Firooz A Sadjadi and Ernest L Hall. Three-dimensional moment invariants. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2) :127–136, 1980.

[SKBM13] Anthony Sorel, Richard Kulpa, Emmanuel Badier, and Franck Multon. Dealing with variability when recognizing user's performance in natural 3d gesture interfaces. *International Journal of Pattern Recognition and Artificial Intelligence*, 27(08) :1350023, 2013.

[SKS12] Min Sun, Pushmeet Kohli, and Jamie Shotton. Conditional regression forests for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3394–3401. IEEE, 2012.

[SMMN12] Loren Arthur Schwarz, Artashes Mkhitaryan, Diana Mateus, and Nassir Navab. Human skeleton tracking from depth data using geodesic distances and optical flow. *Image and Vision Computing*, 30(3) :217–226, 2012.

[Sor12] Anthony Sorel. *Gestion de la variabilité morphologique pour la reconnaissance de gestes naturels à partir de données 3D*. PhD thesis, Université Rennes 2, 2012.

[SPSS11] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Human activity detection from rgbd images. *plan, activity, and intent recognition*, 64, 2011.

[SPSS12] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena. Unstructured human activity detection from rgbd images. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 842–849. IEEE, 2012.

[SSB14] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[SSK+13] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1) :116–124, 2013.

[SSS05] Yaser Sheikh, Mumtaz Sheikh, and Mubarak Shah. Exploring the space of a human action. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 144–149. IEEE, 2005.

[STHES15] Amr Sharaf, Marwan Torki, Mohamed E Hussein, and Motaz El-Saban. Real-time multi-scale action detection from 3d skeleton data. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2015*, pages 998–1005, 2015.

[SWDS15] Rim Slama, Hazem Wannous, Mohamed Daoudi, and Anuj Srivastava. Accurate 3d action recognition using learning on the grassmann manifold. *Pattern Recognition*, 48(2) :556–567, 2015.

[TKEF14] Ilias Theodorakopoulos, Dimitris Kastaniotis, George Economou, and Spiros Fotopoulos. Pose-based human action recognition via sparse representation in dissimilarity space. *Journal of Visual Communication and Image Representation*, 25(1) :12–23, 2014.

[VAC14] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2014, 2014.

[VFML17] Thales Vieira, Romain Faugeroux, Dimas Martínez, and Thomas Lewiner. Online human moves recognition through discriminative key poses

and speed-aware action graphs. *Machine Vision and Applications*, 28(1-2) :185–200, 2017.

[Vic18] *Vicon*, Available at `http://www.vicon.com/products/software/nexus`. 2018 (accessed April 23, 2018).

[Vit67] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2) :260–269, 1967.

[Wik18] Wikipedia. *Réseau de neurones récurrents*, Available at `https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_r%C3%A9currents`. 2018 (accessed April 23, 2018).

[WLW14] Jiang Wang, Zicheng Liu, and Ying Wu. Learning actionlet ensemble for 3d human action recognition. In *Human Action Recognition with Depth Cameras*, pages 11–40. Springer, 2014.

[WLWY12] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1290–1297. IEEE, 2012.

[WWCL13] Yeh-Kuang Wu, Hui-Chun Wang, Liung-Chun Chang, and Ke-Chun Li. Using hmms and depth information for signer-independent sign language recognition. In *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, pages 79–86. Springer, 2013.

[WWL07] Jacob O Wobbrock, Andrew D Wilson, and Yang Li. Gestures without libraries, toolkits or training : a one-dollar recognizer for user interface prototypes. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 159–168. ACM, 2007.

[WWY13] Chunyu Wang, Yizhou Wang, and Alan L Yuille. An approach to pose-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 915–922, 2013.

[WYW+12a] Youwen Wang, Cheng Yang, Xiaoyu Wu, Shengmiao Xu, and Hui Li. Kinect based dynamic hand gesture recognition algorithm research. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*, volume 1, pages 274–279. IEEE, 2012.

[WYW+12b] Xiaoyu Wu, Cheng Yang, Youwen Wang, Hui Li, and Shengmiao Xu. An intelligent interactive system based on hand gesture recognition algorithm and kinect. In *Computational Intelligence and Design (ISCID), 2012 Fifth International Symposium on*, volume 2, pages 294–298. IEEE, 2012.

[WZZZ13a] Ping Wei, Yibiao Zhao, Nanning Zheng, and Song-Chun Zhu. Modeling 4d human-object interactions for event and object recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3272–3279, 2013.

[WZZZ13b] Ping Wei, Nanning Zheng, Yibiao Zhao, and Song-Chun Zhu. Concurrent action detection with structural prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3136–3143, 2013.

[XCA12] Lu Xia, Chia-Chih Chen, and JK Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 20–27. IEEE, 2012.

[XCL+12] Dan Xu, Yen-Lun Chen, Chuan Lin, Xin Kong, and Xinyu Wu. Real-time dynamic gesture recognition system based on depth perception for robot navigation. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 689–694. IEEE, 2012.

[YJLSHDY15] Ho Yub Jung, Soochahn Lee, Yong Seok Heo, and Il Dong Yun. Random tree walk toward instantaneous 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2467–2474, 2015.

[YLY14] Gang Yu, Zicheng Liu, and Junsong Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. In *Asian Conference on Computer Vision*, pages 50–65. Springer, 2014.

[YT12] Xiaodong Yang and Ying Li Tian. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on*, pages 14–19. IEEE, 2012.

[YT14] Xiaodong Yang and YingLi Tian. Effective 3d action recognition using eigenjoints. *Journal of Visual Communication and Image Representation*, 25(1) :2–11, 2014.

[ZCG13] Yu Zhu, Wenbin Chen, and Guodong Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 486–491, 2013, 2013.

[Zha12] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2) :4–10, 2012.

[ZLP+13] Xin Zhao, Xue Li, Chaoyi Pang, Xiaofeng Zhu, and Quan Z Sheng. Online human gesture recognition from motion data streams. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 23–32. ACM, 2013.

[ZLP+14] Xin Zhao, Xue Li, Chaoyi Pang, Quan Z Sheng, Sen Wang, and Mao Ye. Structured streaming skeleton–a new feature for online human gesture recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 11(1s) :22, 2014.

[ZLS13] Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. The moving pose : An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2752–2759, 2013.

[ZLX+16] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, Xiaohui Xie, et al. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In *AAAI*, volume 2, page 8, 2016.

[ZLX17] Songyang Zhang, Xiaoming Liu, and Jun Xiao. On geometric features for skeleton-based action recognition using multilayer lstm networks. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 148–157. IEEE, 2017.

[ZP12] Hong-Min Zhu and Chi-Man Pun. Real-time hand gesture recognition from depth image sequences. In *Computer Graphics, Imaging and Visualization (CGIV), 2012 Ninth International Conference on*, pages 49–52. IEEE, 2012.

[ZP15] Hao Zhang and Lynne E Parker. Bio-inspired predictive orientation decomposition of skeleton trajectories for real-time human activity predic-

tion. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3053–3060. IEEE, 2015.

[ZT12] Chenyang Zhang and Yingli Tian. Rgb-d camera-based daily living activity recognition. *Journal of Computer Vision and Image Processing*, 2(4) :12, 2012.

[ZWZ13] Weijia Zou, Baoyuan Wang, and Rui Zhang. Human action recognition by mining discriminative segment with novel skeleton joint feature. In *Pacific-Rim Conference on Multimedia*, pages 517–527. Springer, 2013.

[ZYX+18] Songyang Zhang, Yang Yang, Jun Xiao, Xiaoming Liu, Yi Yang, Di Xie, and Yueting Zhuang. Fusing geometric features for skeleton-based action recognition using multilayer lstm networks. *IEEE Transactions on Multimedia*, 2018.

Table des figures

Liste des tableaux

Titre : Reconnaissance en-ligne d'actions 3D par l'analyse des trajectoires du squelette humain.

Mots clés : trajectoires squelettiques, reconnaissance d'actions, interaction en-ligne

Résumé : L'objectif de cette thèse est de concevoir une approche transparente originale apte à détecter en temps-réel l'occurrence d'une action (geste 3D), dans un flot non segmenté et idéalement le plus tôt possible. Ces travaux s'inscrivent dans une collaboration entre deux équipes de l'IRISA-Inria de Rennes, à savoir Intuidoc et MimeTIC. En profitant de la complémentarité des savoir-faire des deux équipes de recherche, nous proposons de reconsidérer les besoins et les difficultés rencontrées pour modéliser, reconnaître et détecter une action 3D en proposant de nouvelles solutions à la lumière des avancées réalisées en termes de modélisation de gestes manuscrits 2D.

Les contributions de cette thèse sont regroupées en trois parties principales. Dans la première partie, nous proposons une nouvelle approche pour modéliser et reconnaître une action pré-segmentée. En effet, il est d'abord nécessaire de développer une représentation à même de caractériser le plus finement possible une action donnée pour en faciliter la reconnaissance.

Dans la deuxième partie, nous introduisons une approche permettant de reconnaître une action dans un flot non segmenté : sans connaissance a priori du début ni de la fin de l'action. Enfin, dans la troisième partie, nous étendons cette dernière approche pour la caractérisation précoce d'une action avec très peu de d'information (c.-à-d. le début de l'action).

Pour chacune de ces trois problématiques, nous avons identifié explicitement les difficultés à considérer afin d'en effectuer une description complète pour permettre de concevoir des solutions ciblées pour chacune d'elles. Les résultats expérimentaux obtenus sur différents benchmarks d'actions attestent de la validité de notre démarche. En outre, à travers des coopérations ayant eu lieu au cours de la thèse, les approches développées ont été déployées dans trois applications, dont des applications en animation et en reconnaissance de gestes dynamiques de la main.

Title : Online 3D actions recognition by analyzing the trajectories of human's skeleton

Keywords : skeletal trajectories, actions recognition, online interaction

Abstract : The objective of this thesis is to design an original transparent approach able to detect in real time the occurrence of an action (3D gesture), in an unsegmented flow and ideally as early as possible. This work is part of a collaboration between two IRISA-Inria teams in Rennes, namely Intuidoc and MimeTIC. By taking advantage of the complementary expertise of the two research teams, we propose to reconsider the needs and difficulties encountered to model, recognize and detect a 3D action by proposing new solutions in the light of the advances made in terms of 2D handwriting modeling.

The contributions of this thesis are grouped into three main parts. In the first part, we propose a new approach to model and recognize a pre-segmented action. Indeed, it is first necessary to develop a representation able to characterize as finely as possible a given action to facilitate recognition.

In the second part, we introduce an approach to recognize an action in an unsegmented flow: without prior knowledge of the beginning or the end of the action. Finally, in the third part, we extend this last approach for the early characterization of an action with very little information (ie the beginning of the action).

For each of these three issues, we have explicitly identified the difficulties to be considered in order to make a complete description of them so that we can design targeted solutions for each of them. The experimental results obtained on different benchmarks of actions attest to the validity of our approach. In addition, through collaborations that took place during the thesis, the developed approaches were deployed in three applications, including applications in animation and in dynamic hand gestures recognition.