Vers une reconnaissance en ligne d'actions à partir de caméras RGB-D

Enjie Ghorbel, Rémi Boutteau, Jacques Boonaert, Xavier Savatier, Stéphane Lecoeuche

Vers une reconnaissance en ligne d'actions à partir de caméras RGB-D

E. Ghorbel[1,2], R. Boutteau [1], J. Boonaert [2], X. Savatier [1], S. Lecoeuche [2]

[1] Institut de Recherche en Systémes Eléctroniques Embarqués (IRSEEM), ESIGELEC, Université de Rouen, EA4353, Rouen
[2] Unité de Recherche en Informatique et Automatique (URIA), Mines Douai, Douai

[1] prenom.nom@esigelec.fr
[2] prenom.nom@mines-douai.fr

Résumé

Classiquement, l'évaluation des méthodes de reconnaissance d'actions se fait à l'aide d'un critère principal : le taux de reconnaissance. Cependant, une méthode dont le temps de calcul est important n'est utilisable que pour un nombre très restreint d'applications. Dans ce papier, nous présentons une nouvelle approche pour la reconnaissance d'actions à partir de caméras RGB-D, laquelle permet d'aboutir à un taux de reconnaissance intéressant tout en minimisant le coût en temps de calcul. Ainsi, un nouveau descripteur basé sur l'interpolation par splines cubiques des grandeurs cinématiques du squelette est proposé. Dans l'optique d'étendre notre méthode aux scénarios en ligne par l'utilisation d'une fenêtre glissante, nous menons une étude expérimentale qui nous permettra de l'évaluer sur des actions incomplètes. Ainsi, nous présentons une nouvelle base de données d'actions incomplètes IncompleteMSRAction3D, générée à partir de la base de référence MSRAction3D.

Mots Clefs

Reconnaissance d'actions incomplètes, caméras RGB-D, latence calculatoire.

Abstract

Generally, the evaluation of action recognition method is done thanks to a main criterion : the accuracy of recognition. However, if the computational latency of a given method is too important, it becomes difficult to apply it in a wide range of applications. In this paper, we introduce a novel approach of action recognition based on RGB-D cameras which presents a good accuracy and a low execution time. Thus, a new descriptor based on the cubic spline interpolation of kinematic values is proposed. To evaluate the eventual applicability of our approach in online scenarios (unsegmented videos) with a sliding window approach, we propose to train a model of classification using incomplete actions which are randomly generated from the MSRAction3D benchmark.

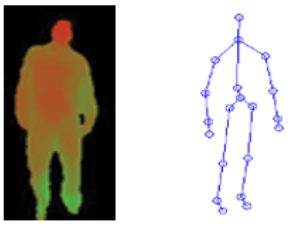

FIGURE 1 – Exemple d'image de profondeur (à gauche) et de squelette (à droite)

Keywords

Recognition of incomplete actions, RGB-D cameras, computational latency.

1 Introduction

La reconnaissance d'actions suscite, de plus en plus, l'intérêt de la communauté scientifique du domaine en raison de son large champ applicatif. Permettant d'obtenir des séquences d'images RGB (*Red Green Blue*), les caméras classiques ont représenté les systèmes d'acquisition sur lesquels se sont basées les méthodes les plus populaires[1]. Néanmoins, il est important de noter que ces caméras présentent certaines limitations à savoir : la sensibilité aux changements d'illumination, aux occultations, aux changements d'orientation et à la segmentation.

Durant cette dernière décennie, les caméras RGB-D (*Red Green Blue Depth*) ont été introduites sur le marché. Elles permettent de fournir non seulement des séquences d'images RGB, mais également des séquences d'images de profondeur. En complément, Shotton et al. [19] ont proposé un algorithme permettant d'extraire une séquence de squelettes assez robuste à partir des cartes de profondeur (environ 45 ms par image d'après [17]). Les chercheurs ont alors commencé à introduire de nouveaux descripteurs en se basant sur l'une de ces deux modalités (cartes de

profondeur et séquences de squelettes) (Figure 1). Ainsi, un grand nombre de publications récentes proposent des descripteurs très performants en termes de taux de reconnaissance. Toutefois, les auteurs ne s'intéressent généralement pas à la performance de ces descripteurs en termes de temps de calcul [5][9][10][8]. Pourtant, la rapidité de calcul est un facteur essentiel pour les applications réclamant une réponse instantanée, qu'elles se trouvent dans une configuration hors-ligne (la rééducation, le coaching, *etc.*) ou en ligne (les actions ne sont pas pré-segmentées). D'après [16], la solution idéale serait de trouver un compromis entre un taux reconnaissance acceptable et un temps de calcul faible.

Ainsi, une nouvelle méthode de reconnaissance d'actions à partir de caméras de profondeur est proposée afin de concilier précision et rapidité. Le descripteur de mouvement proposé est construit grâce à l'interpolation des grandeurs cinématiques du squelette. L'évaluation de cette méthode (en termes de temps d'exécution et taux de reconnaissance) est basée sur une étude comparative sur la base de données MSRAction3D [12]. Dans l'objectif d'étendre ultérieurement cette approche à la reconnaissance en ligne, nous avons mené des premières expérimentations sur des actions incomplètes. Pour ce faire, une nouvelle base de données que nous avons appelée IncompleteMSRAction3D a été générée à partir de la base de données MSRAction3D.

Cet article est organisé comme suit : La Section 2 synthétise l'état de l'art de la reconnaissance d'actions basée sur les caméras RGB-D, puis, la Section 3 présente la méthode développée en détaillant le descripteur proposé, tout en expliquant la procédure de classification suivie. La Section 4 présente les bases de données utilisées, les expérimentations ainsi que les résultats. Finalement, la Section 5 conclut ce travail et évoque quelques pistes que nous espérons exploiter dans des travaux futurs.

2 Travaux antérieurs

Dans cette section, nous décrivons de manière succincte les principales méthodes de reconnaissance d'actions à partir des caméras RGB-D. Ces méthodes sont souvent catégorisées selon le type de descripteur utilisé. Il s'agit principalement des deux types suivants : les descripteurs de profondeur et les descripteurs basés sur le squelette.

2.1 Les descripteurs basés sur la profondeur

Ce type de descripteur est extrait à partir des cartes de profondeur fournies par des caméras RGB-D. Une première génération de descripteurs s'inspire des méthodes initialement développées pour les images RGB. Nous citons par exemple les travaux de Xia et al. [5] qui introduisent un algorithme permettant de détecter des STIPs (*Spatio-Temporal Interest Points*) sur des cartes de profondeur. Une extension 4D du descripteur cuboïde [6] est également proposée dans cet article. Klaser et al. [7] proposent, quant à eux, l'extension du HOG [20] classique au HOG3D. De même, Ohn-Bar et al. [8] étendent le HOG au HOG2, mettant en jeu deux histogrammes (l'un spatial et l'autre temporel). Toutefois, il a été récemment démontré que ces méthodes ne peuvent pas se montrer optimales dans le cas des images de profondeur, qui possèdent une structure différente [9].

De nouvelles approches, modélisant le corps humain et son mouvement par une hypersurface de $\mathbb{R}^4$, sont alors apparues. Plusieurs travaux se basent sur l'exploitation des normales de cette hypersurface. A titre d'exemple, Oreifeij et al [9] construisent un histogramme de normales 4D (HON4D). Se référant également aux normales 4D, Yang et al. [10] proposent de calculer ce qu'ils appellent des SNV (Super Normal Vector) obtenus grâce au calcul de polynormales autour de zones bien spécifiques.

Les descripteurs de profondeur se sont généralement révélés plus performants que d'autres descripteurs basés sur le squelette ou sur les images RGB. Ceci pourrait s'expliquer par le fait que les cartes de profondeurs seraient plus robustes au bruit et aux occultations. Cependant, vu la grande dimension des images de profondeur, le temps de calcul des descripteurs peut s'avérer assez important [22, 23]. Nous reviendrons sur ce point en nous appuyant sur des preuves expérimentales dans la Section 4.

2.2 Les descripteurs basés sur le squelette

Grâce aux bibliothèques comme OpenNI ou KinectSDK, il est devenu possible d'extraire des séquences de squelette assez précises de manière quasi-instantanée (45 ms par image d'après). Chaque squelette est généralement constitué de 15 ou 20 articulations (cela dépend de l'algorithme utilisé). Dû à sa faible dimension et à la popularité de la représentation par squelette dans un grand nombre d'études biomécaniques, une grande variété de travaux s'est consacrée à cette modalité.

Un des premiers travaux proposant l'utilisation des articulations pour décrire le mouvement humain a été introduit par Li et al. [12]. L'idée consiste à modéliser un graphe d'actions en utilisant des sacs de points 3D. Un autre papier décrit l'action en construisant des Histogrammes 3D d'articulations orientés (HOJ3D) [11].

Pour réduire l'effet de la variabilité anthropométrique, beaucoup d'auteurs préfèrent utiliser les positions d'articulations relatives. Par exemple, Yang et al. [13] utilisent les articulations propres (*EigenJoints*) prenant ainsi en compte les distances spatiales et temporelles entre les articulations.

Des descripteurs beaucoup plus performants ont été récemment introduits mettant en œuvre des notions biomécaniques. En effet, le corps humain peut être représenté par une séquence de référentiels locaux qui seraient chacun relié à un segment du squelette. Dans [14], les matrices de transformation entre les différents segments sont exprimées dans $SE(3)$. Chaque pose est associée à un point de $SE(3)^n$, où n représente le nombre d'articulations. L'idée majeure de cette méthode est d'interpoler ces points et de calculer la distance entre la courbe interpolée et les autres exemples. Enfin, Zanfir et al. [15] proposent l'utilisation

d'un descripteur concaténant les informations de position, de vitesse et d'accélération des articulations. Il associe à chaque grandeur un poids empirique.
Inspirés par ces deux derniers travaux, nous proposons de calculer la position, la vitesse et l'accélération de chaque articulation et d'interpoler ces valeurs physiques à l'aide de splines cubiques afin d'obtenir des descripteurs compacts de même taille. Cette méthode permet alors d'obtenir un bon taux de reconnaissance et un faible coût calculatoire. La performance d'un grand nombre de descripteurs basés sur les modalités profondeur et squelette a été montrée dans la littérature en rapportant uniquement le taux de reconnaissance. Cependant, très peu d'articles ont évalué leurs descripteurs en termes de temps de calcul. Comme évoqué plus tôt, les descripteurs basés sur la profondeur sont généralement très coûteux à calculer vu la grande dimension des cartes de profondeur. Les descripteurs de squelette sont généralement assez rapides à calculer. Toutefois, le temps de calcul de ces descripteurs est parfois augmenté par des étapes d'optimisation, des pré-traitements complexes ou de calculs numériques coûteux. Les composantes de cette approche ont été choisies de sorte à éviter les calculs trop coûteux (tels que le passage du groupe de Lie à l'algèbre de Lie et l'interpolation dans $SE(3)^n$ dans [14]) ou les choix empiriques (tels que les différents poids associés à chaque grandeur physique dans [15]).

3 Méthodes

La reconnaissance d'actions est souvent schématisée comme la succession de deux processus : la construction d'un descripteur de mouvement suivi de la classification. Dans ce qui suit, nous commençons par présenter le nouveau descripteur calculé à partir de l'information du squelette. Puis, nous réservons une sous-section à l'étape de classification et expliquons le choix du classifieur de type SVM (Machine à Vecteurs Supports) linéaire. La Figure 2 représente une vue globale de la méthode proposée dans cet article.

3.1 Descripteur d'actions basés sur la cinématique du squelette

Dans cette partie, nous détaillons la construction du nouveau descripteur proposé afin de satisfaire un compromis entre un temps de calcul faible et un bon taux de reconnaissance. Une première étape de pré-traitement a été mise en place afin de pallier la variabilité spatiale. Puis, les caractéristiques cinématiques (position, vitesse, accélération) des articulations ont été calculées à partir de chaque squelette. En raison de la longueur des séquences très variable, une étape de ré-échantillonnage est nécessaire. Ceci est réalisé grâce à l'interpolation par splines cubiques des caractéristiques cinématiques suivie d'un échantillonnage périodique proportionnel à la longueur de la séquence. Dans cet article, nous supposons que l'acquisition de vidéos est soumise à des conditions particulières : 1) Le sujet fait face à la caméra. 2) Chaque vidéo contient au plus un seul sujet 3) Les vidéos ne contiennent pas d'interactions homme-objet.
Pré-traitement des articulations du squelette : Chaque action est représentée par une séquence de *M* squelettes et chaque squelette est composé par *n* articulations. Les caméras RGB-D retournent la position $p_{\mathrm{i}}(t)$ de chaque articulation *i* à l'instant *t* dans le repère monde par l'utilisation de bibliothèques telles que OPENI ou KinectSDK :

$$p_{\mathrm{i}}(t) = [x_{\mathrm{i}}(t), y_{\mathrm{i}}(t), z_{\mathrm{i}}(t)]^T \quad (1)$$

De ce fait, une action peut être assimilée à une série temporelle (2).

$$p(t) = [p_1(t), p_2(t), ..., p_{\mathrm{n}}(t)]^T \quad (2)$$

En biomécanique, il est très fréquent de prendre l'articulation de la hanche comme origine du référentiel lié au corps humain (3).

$$p^{\mathrm{hip}}(t) = [p_1(t) - p_{\mathrm{hip}}, p_2(t) - p_{\mathrm{hip}}, ..., p_{\mathrm{n}}(t) - p_{\mathrm{hip}}]^T \quad (3)$$

Le pré-traitement mis au point est réalisé dans le but de réduire la sensibilité de notre descripteur aux variations anthropométriques. En effet, la longueur des segments du squelette varie d'un individu à un autre. Par exemple, considérons deux sujets 1 et 2 tels que *S1(i)* et *S2(i)* représentent les tailles respectives de leurs segments *i*, et tels que *S1(j)* et *S2(j)* soient égales aux tailles respectives de leurs segments *j*. On peut aisément remarquer que généralement *S1(i)/S2(i)* diffère de *S1(j)/S2(j)*. C'est ainsi que nous proposons de normaliser le squelette au sens anthropométrique.
Inspirés par l'idée de normalisation développée dans [15], nous proposons de normaliser les segments au sens euclidien, au lieu de faire l'apprentissage d'un squelette moyen et de contraindre tous les squelettes de la base de données à avoir les même dimensions que le squelette modèle. Ceci revient à normaliser de manière itérative les segments (diviser par la norme afin d'obtenir des segments unitaires) en commençant par une racine (articulation de la hanche) et en passant successivement aux segments voisins. Cette approche permet de conserver la forme du squelette. Ce pré-traitement nous permet alors d'obtenir la position des articulations normalisées p^{norm} (4) d'un point de vue anthropométrique.

$$p^{\mathrm{norm}} = normalisation(p^{\mathrm{hip}}) \quad (4)$$

Calcul des caractéristiques cinématiques : Toujours dans une démarche biomécanique, nous supposons que le mouvement est caractérisé par la position, la vitesse et l'accélération des articulations. Parmi ces trois grandeurs physiques, nous ne disposons que de l'information discrète de la position des articulations. Nous utilisons cette même information pour calculer le reste des caractéristiques cinématiques à savoir la vitesse (5) et l'accélération (6).

$$V(t) = p^{\mathrm{norm}}(t+1) - p^{\mathrm{norm}}(t-1) \quad (5)$$

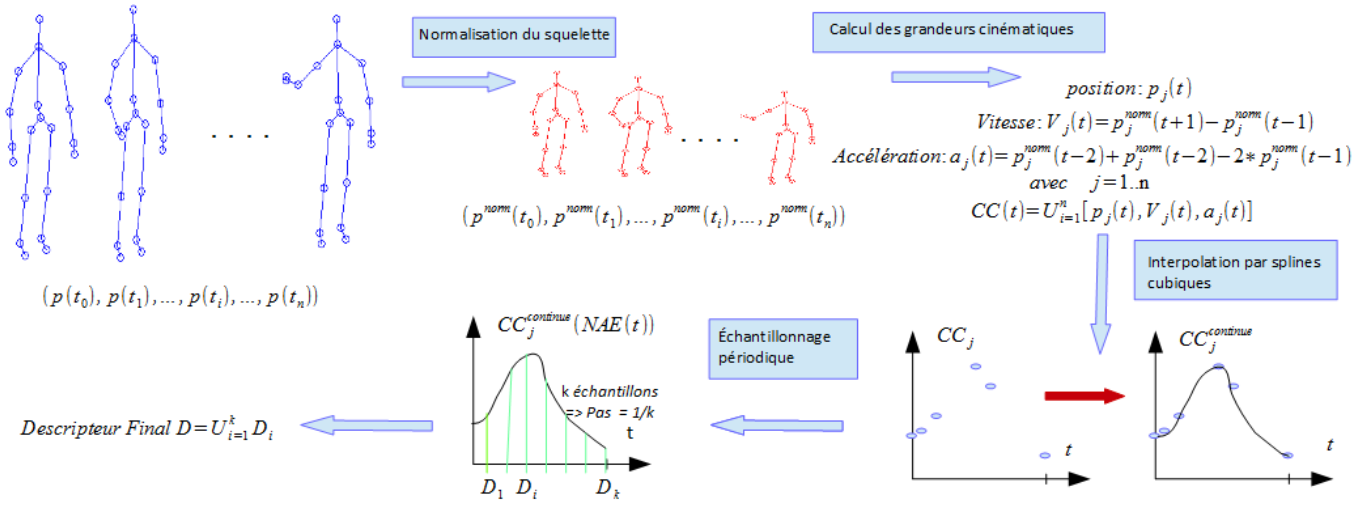

FIGURE 2 – Vue générale de notre méthode

$$a(t) = p^{\text{norm}}(t+2) + p^{\text{norm}}(t-2) - 2 \times p^{\text{norm}}(t) \quad (6)$$

Les caractéristiques cinématiques $CC(t)$ sont alors exprimées par la concaténation des trois grandeurs :

$$CC(t) = [p^{\text{norm}}(t), V(t), a(t)]^T \quad (7)$$

Ainsi, pour un instant t bien déterminé, la dimension de $CC(t)$ est égale à $9 \times n$.

La longueur des vidéos varie d'une instance à une autre. Pour cela, il est nécessaire d'inclure une étape de ré-échantillonnage qui va permettre d'obtenir des vecteurs de même taille avant de commencer l'étape de classification. Dans ce qui suit, nous décrivons l'étape de ré-échantillonnage constituée de 2 sous-étapes : L'interpolation par splines cubiques des caractéristiques cinématiques et l'échantillonnage périodique des courbes obtenues.

Interpolation par spline cubique : Considérant que le mouvement humain est continu dans le temps, nous admettons également que la position, la vitesse et l'accélération représentent des variables continues dans le temps. C'est ainsi que nous proposons d'interpoler dans le temps par des splines cubiques chacune des composantes du vecteur contenant les caractéristiques cinématiques $CC(t)$ afin d'obtenir des caractéristiques continues $CC^{continue}(t)$. Nous notons par $CC_j(t)$ la j-ème composante du vecteur de composantes cinématiques $CC(t)$ à l'instant t. De même, $CC_j^{continue}$ représente la courbe résultant de l'interpolation dans le temps des j-ème composantes de CC_j. L'interpolation par splines cubiques est une technique d'interpolation très connue. Elle permet de connecter un ensemble de points discrets de manière cohérente. Ce type d'interpolation présente trois avantages : sa simplicité, sa rapidité et le fait que celle-ci permette d'obtenir des courbes réalistes en limitant le nombre d'oscillations (un point d'inflexion vu le degré trois du polynôme). L'équation (8) décrit cette interpolation où t_c représente la variable de temps continue tel que $t_c \in [0, M]$.

$$CC_j^{continue}(t_c) = spline(CC_j(t))_{1 \leqslant t \leqslant M} \quad avec\ j = 1...9 \times n \quad (8)$$

Échantillonnage :
Finalement, afin d'obtenir des descripteurs de même taille quelle que soit la taille de la séquence d'entrée, chaque vecteur de caractéristiques continues est échantillonné proportionnellement à la taille de la vidéo de sorte à obtenir pour chaque séquence k échantillons. Le descripteur final D est donc obtenu grâce à l'équation (9). M représente la longueur de la vidéo.

$$D = \cup_{j=1..k} CC^{continue}(j \times M/k) \quad (9)$$

3.2 Reconnaissance d'actions via une Machine à Vecteurs de Support linéaire

Nous proposons d'utiliser un classifieur de type machine à vecteurs de support (SVM) multi-classe et linéaire [2]. Comme il a été montré dans plusieurs papiers [14][9], ce classifieur est assez intéressant dans le cas de la reconnaissance d'actions via les caméras RGB-D. De plus, les classifieurs linéaires possèdent un coût calculatoire très faible [3]. Ainsi, pour le cas hors ligne, il suffit d'extraire sur chaque vidéo (correspondant à une action) le descripteur proposé et de réaliser l'étape de reconnaissance grâce au modèle appris par la SVM linéaire. Pour réaliser cette étape, nous utilisons la bibliothèque *libLinear* [18].

4 Expérimentations et Résultats

Dans cette partie, nous présentons les expérimentations menées ainsi que les résultats obtenus. Dans un premier temps nous comparons notre méthode à d'autres méthodes

sur la base de référence MSRAction3D en termes de taux de reconnaissance et de temps d'exécution. Dans un second temps, nous proposons de présenter nos premiers résultats sur la base de données d'actions incomplètes IncompleteMSRAction3D générée à partir de MSRAction3D.

4.1 Reconnaissance d'actions sur MSRAction3D

La base de données MSRAction3D : Afin de positionner notre travail par rapport à l'état de l'art, nous proposons de tester notre approche sur la base de données MSRAction3D qui représente la base de données de référence dans le domaine de la reconnaissance d'actions via les caméras RGB-D. Elle rassemble 20 types d'actions humaines. Chaque action est réalisée deux ou trois fois par dix sujets différents. Ainsi, cette base comporte en tout 567 actions segmentées. Comme dans [5], nous conservons 557 actions en éliminant 10 actions très biaisées. MSRAction3D comporte deux types deux modalités : des cartes de profondeur et des séquences de squelette extraites par l'utilisation de l'algorithme de Shotton et al. [19]. La difficulté de cette base de données réside dans le fait que celle-ci contient des actions très similaires.

Conditions et paramètres d'expérimentation : Comme évoqué dans [21], beaucoup de travaux comparent leur méthodes à d'autres sur la base de données MSRAction3D sans respecter les mêmes conditions et paramètres d'expérimentation [8][5]. Afin d'obtenir une comparaison fiable, nous proposons de rassembler les codes disponibles des méthodes les plus performantes et les plus récentes, puis de refaire l'expérimentation dans les mêmes conditions et sur le même matériel. Les descripteurs rassemblés peuvent être divisés en deux catégories. La première catégorie rassemble les descripteurs de profondeur et inclut HOG2 [8], SNV [10] et HON4D [9], tandis que la seconde catégorie rassemble les descripteurs basés sur la modalité squelette où les caractéristiques ont été re-paramétrées grâce à l'utilisation du DTW (*Dynamic Time Warping*). Cette dernière catégorie comprend les JP [14] (*Joint Position*), les RJP [14] (*Relative Joint Position*), les Q [14] (*Quaternions*) et les LARP [14] (*Lie Algebra Relative Position*). Pour une expérimentation fiable, nous respectons le protocole le plus courant qui répartit la base de données MSRAction3D en trois sous-bases comportant chacune 8 types d'actions : AS1, AS2 et AS3 [12]. Les deux premiers sous-ensembles regroupent des actions très similaires tandis que AS3 regroupe des actions complexes. L'apprentissage et le test sont alors faits indépendamment sur chaque sous-base, puis le taux de reconnaissance global est obtenu en moyennant ceux calculés sur chaque sous-base. Il est également important de préciser que les données produites par les sujets 1,3,5,7,9 ont été utilisées pour l'apprentissage lorsque les données produites par le reste des sujets ont été utilisées pour le test.

Critères d'évaluation : Comme évoqué précédemment, les méthodes antérieures se comparent généralement entre elles en se basant sur l'unique critère de taux de reconnaissance. Pour grand nombre d'applications, un temps d'exécution faible représente également une caractéristique nécessaire. Dans cette logique, nous proposons d'ajouter un second critère d'évaluation à la reconnaissance d'actions : le temps moyen d'exécution par descripteur (T.E.). Dans toutes les méthodes testées, la classification est réalisée via un SVM linéaire dont le coût calculatoire est négligeable comparée au calcul du descripteur.

Descripteur	AS1(%)	AS2(%)	AS3(%)	T.R. (%)	T.E. (s)
HOG2 [8]	90.47	84.82	**98.20**	91.16	**6.44**
HON4D [9]	94.28	91.71	**98.20**	94.47	27.33
SNV [10]	**95.25**	**94.69**	96.43	**95.46**	146.57
JP [14]	82.86	68.75	83.73	78.44	0.58
RJP [14]	81.90	71.43	88.29	80.53	2.15
Q [14]	66.67	59.82	71.48	67.99	1.33
LARP [14]	83.81	84.82	92.73	87.14	17.61
Notre approche	**84.11**	**87.5**	**97.3**	**89.64**	**0.092**

TABLE 1 – Taux de reconnaissance (T.R.) et temps d'exécution moyen par descripteur (T.E.) sur MSRAction3D : AS1, AS2 and AS3 représentent les trois groupes d'actions proposées dans le protocole d'expérimentation [12]

Résultats : Les valeurs de taux de reconnaissance et de temps d'exécution moyen par descripteur (des méthodes antérieures et de notre méthode) sont rapportées dans le tableau 1. La supposition qui stipule que les descripteurs de profondeur seraient plus précis mais plus lents à calculer est confirmée par les résultats obtenus. D'un autre côté, il est clair que notre méthode possède le descripteur le plus rapide à calculer avec un temps d'exécution moyen par descripteur égal à 9,2 ms. De plus, comparé aux descripteurs basés squelette (JP, RJP, Q, LARP), le descripteur proposé est le plus performant en termes de taux de reconnaissance.

Intérêt de la normalisation du squelette : La normalisation du squelette associée à notre méthode permet d'améliorer le taux de reconnaissance de 5% sur la base de données. Sans normalisation, le taux de reconnaissance n'atteint que 89.64%.

Intérêts des grandeurs cinématiques : Dans le tableau 3, nous tentons d'évaluer l'importance de chacune des grandeurs cinématiques utilisées, à savoir la position (P), la vitesse (V) et l'accélération (A). D'après ce tableau, la position représente la grandeur la plus discriminante, suivie de la vitesse, puis de l'accélération. Cela pourrait s'expliquer par l'erreur causée par les dérivations. Toutefois, il est important de noter que le meilleur taux de reconnaissance est obtenu lorsque les trois grandeurs sont utilisées.

4.2 Vers une reconnaissance en ligne d'actions : Tests préliminaires de Reconnaissance d'actions incomplètes

Dans la sous-section précédente, nous avons pu montrer l'intérêt de ce nouveau descripteur pour les applications hors-ligne temps réel. Toutefois, une question se pose : Serait-il possible d'adapter ce même descripteur à la re-

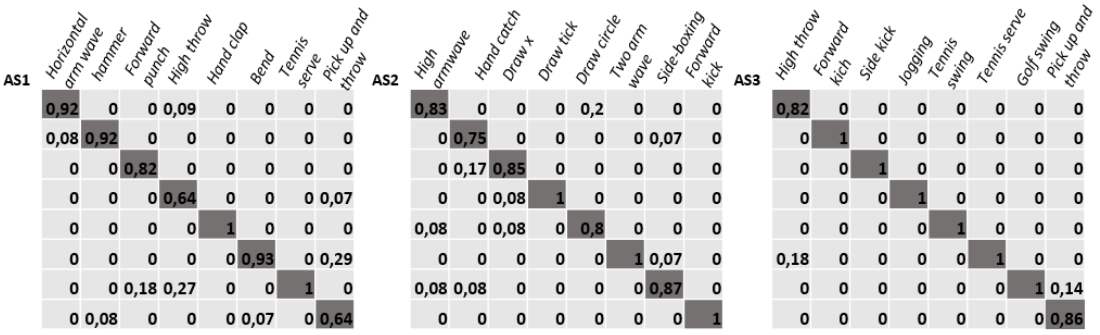

FIGURE 3 – Matrices de confusion sur la base de données MSRAction3D

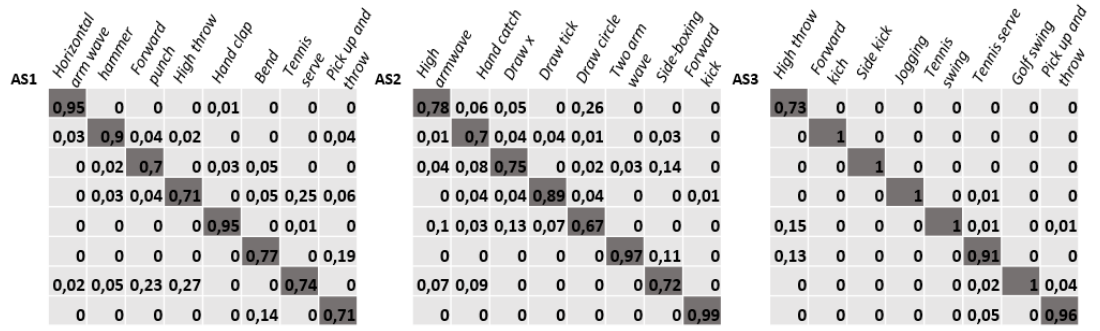

FIGURE 4 – Matrices de confusion sur les vidéos TQV de IncompleteMSRAction3D

Grandeurs utilisées	T.R (%)
P+V+A	**89.64**
P+V	87.28
P	86.63
V	83.90
A	81.47

TABLE 2 – Effet de chaque grandeur cinématique sur le taux de reconnaissance (T.R.) sur la base MSRAction3D

connaissance en ligne avec l'utilisation d'une fenêtre glissante ?

La reconnaissance en ligne comporte deux challenges principaux comparée à la reconnaissance hors-ligne, mis à part le fait que le descripteur se doit d'avoir un coût calculatoire faible :

- Le système doit être capable de reconnaître une action en cours, même si celle-ci n'est pas complète (reconnaissance d'actions incomplètes).
- Le système doit être capable de segmenter les actions, ce qui revient à détecter le début et la fin d'une action.

Dans cet article, nous nous intéresserons tout d'abord à la reconnaissance d'actions incomplètes en évaluant le fonctionnement de notre descripteur sur des actions incomplètes. Ces tests préliminaires pourront éventuellement nous orienter dans des travaux futurs qui s'intéresseront à la problématique de reconnaissance en ligne. Souhaitant étendre notre méthode grâce à une fenêtre glissante, ces tests nous permettrait d'évaluer la reconnaissance en fonction de la taille de cette fenêtre. L'idée serait de répondre à la question suivante : Quel est le temps d'observation minimal nécessaire pour pouvoir reconnaître correctement une action ?

Génération de la base de données IncompleteMSRAction3D : Afin de réaliser ces tests, nous proposons de générer une nouvelle base de données d'actions incomplètes IncompleteMSRAction3D à partir de la base de données MSRAction3D. A partir de chacune des 557 vidéos utilisées de MSRAction3D de taille l , on extrait aléatoirement 10 vidéos de taille $(3 \times l)/4$, 10 vidéos de taille $l/2$ et 10 vidéos de taille $l/4$. Ainsi, nous obtenons une base de données comprenant 16710 actions incomplètes.

Conditions et paramètres d'expérimentation : Pour tester notre descripteur, nous répartissons la base de données IncompeteMSRAction3D en 3 groupes : les actions QV (contenant les vidéos de longueur $l/4$) , les actions DV (contenant les vidéos de longueur $l/2$) et les TQV (contenant les vidéos de longueur $(3 \times l)/4$). L'apprentissage et la reconnaissance sont faits sur chaque groupe séparément. Nous rapportons par la suite le taux de reconnaissance global qui représente la moyenne des taux de reconnaissance sur chacun des groupes (QV, TQV et DV). A l'instar de l'expérimentation sur la base de données MSRAction3D,

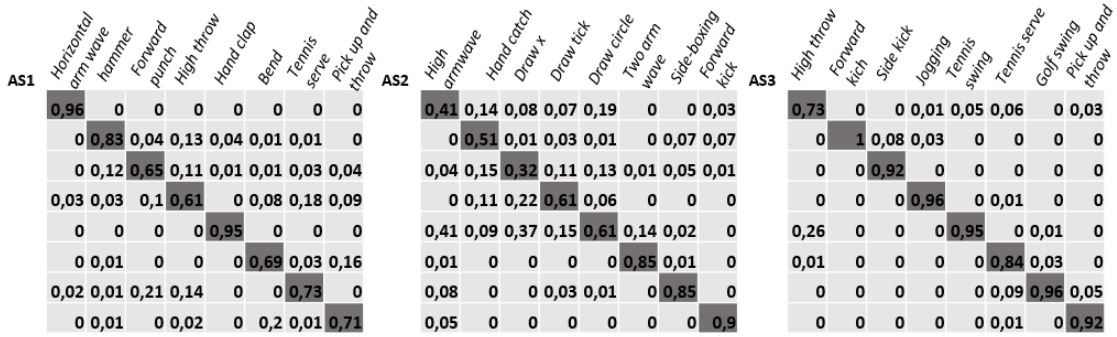

FIGURE 5 – Matrices de confusion sur les vidéos DV de IncompleteMSRAction3D

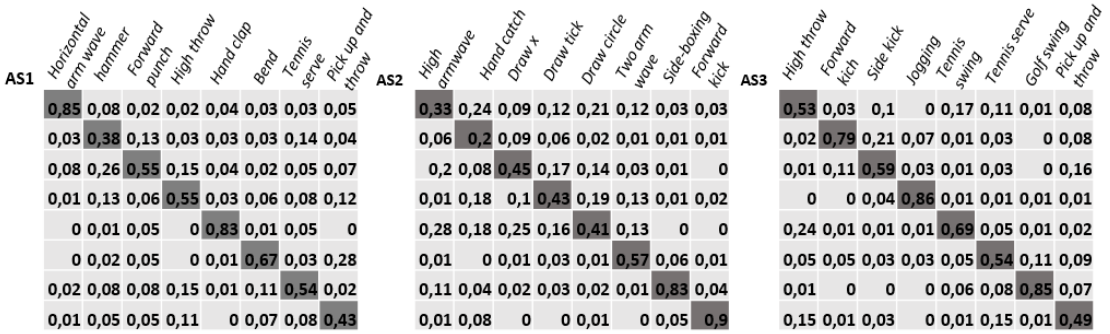

FIGURE 6 – Matrices de confusion sur les vidéos QV de IncompleteMSRAction3D

Descripteur	AS1(%)	AS2(%)	AS3(%)	T.R. (%)
MSR3D	**84.11**	**87.5**	**97.3**	**89.64**
TQV	80.57	81.43	95.59	86.06
DV	77.11	64.64	91.56	77.7
QV	60.93	52.86	68.12	60.63

TABLE 3 – Taux de reconnaissance (T.R.) de notre descripteur sur IncompleteMSRAction3D

les données collectées grâce aux sujets 1,3,5,7,9 sont utilisées pour l'apprentissage du modèle de classification tandis que le reste des données est utilisé pour le test. La répartition de la base de données en trois groupes AS1, AS2 et AS3 est également prise en compte.

Premiers Résultats et Discussion : Le tableau 2, ainsi que les figures 3, 4 et 5 montrent les résultats obtenus sur la base de données IncompleteMSRAction3D. Dans le tableau 2, on observe que plus on raccourcit la taille des vidéos, plus la reconnaissance d'actions est mauvaise. Ce résultat est bien entendu très prévisible. Cependant, en regardant de plus près l'évolution des matrices de confusion, on se rend compte que certaines actions sont beaucoup plus sensibles que d'autres aux raccourcissements des vidéos. A titre d'exemple, ce raccourcissement affecte beaucoup plus l'action *hand catch* que *Golf swing*. En effet, le taux de reconnaissance de *hand catch* passe de 75% sur MSRAction3D à 70% sur TQV, puis à 51% sur DV et pour finir à 20% sur QV. Tandis que pour *Golf swing*, le taux de reconnaissance se détériore moins rapidement avec 100% sur MSRAction3D, 100% sur TQV, 96% sur DV et 85% sur QV. Nous expliquons cela par le fait que l'action *hand catch* ressemble à beaucoup d'autres actions de la base contrairement à l'action *Golf swing* qui se démarque beaucoup plus. Il s'avère alors compliqué de faire la distinction entre deux actions si l'observation se fait sur une fenêtre trop petite. Ainsi, nous reprenons l'idée émise dans [16] : la reconnaissance en ligne dépend du temps de calcul mais également de la latence d'observation qu'il définit par le temps d'observation nécessaire avant de pouvoir prendre une bonne décision. D'après les observations faites à partir des matrices de confusion, nous tendons plutôt à dire que cette latence d'observation dépend beaucoup du type d'action observée, mais également de l'inter-variabilité des classes dans la base de données. De ce fait, il semblerait qu'une approche basée sur des fenêtres glissantes multiples de tailles différentes serait plus adaptée à la reconnaissance en ligne qu'une simple fenêtre glissante.

5 Conclusion et perspectives

Dans ce papier, nous avons proposé une nouvelle méthode de reconnaissance d'actions rapide basée sur des séquences de squelette. Pour cela, un nouveau descripteur de mouve-

ment humain a été introduit mettant en jeu des grandeurs physiques telles que la position, l'accélération et la vitesse que nous avons proposées d'interpoler par des splines cubiques. Ce descripteur se distingue principalement par sa simplicité et par son pouvoir discriminant. En effet, ces qualités ont permis de mettre en œuvre un système de reconnaissance à la fois précis et rapide. Afin de confirmer ces contributions, une comparaison avec les méthodes les plus fiables a été réalisée sur la base de données de référence MSRAction3D. De plus, un second critère d'évaluation a été ajouté au critère de taux de reconnaissance afin de prendre en compte la rapidité de calcul : le temps d'exécution moyen par descripteur. Souhaitant étendre cette méthode dans des travaux futurs à la reconnaissance d'actions en ligne, nous avons testé notre nouvelle approche sur la base de données d'actions incomplètes IncompleteMSRAction3D. Ces premiers tests ont permis d'observer l'effet de la incomplétude de l'action sur la reconnaissance. Cependant, il serait intéressant de comparer sur cette base notre descripteur aux descripteurs récents afin de déterminer lesquels résistent le mieux à la incomplétude de l'information. Nous prévoyons également d'étendre notre méthode à la reconnaissance en ligne grâce à l'utilisation d'une fenêtre glissante (ou de plusieurs fenêtres) et de tester cette extension sur des bases de données non segmentées telles que MSRC-12.

Références

[1] Poppe, R. A survey on vision-based human action recognition. *Image and vision computing*, 28(6), 976-990, 2010.

[2] C. Cortes. and V. Vapnik. Support-vector networks. In *Machine learning*, 20(3), 273-297, 1995.

[3] S.R. Fanello, I. Gori,G. Metta, F. Odone. Keep it simple and sparse : Real-time action recognition. In *The Journal of Machine Learning Research*, 2013, 14(1), 2617-2640.

[4] J.K. Aggarwal. and M.S. Ryoo. Human activity analysis : A review. In *ACM Computing Surveys*, 43(3) :16 :1-16 :43,2011.

[5] L. Xia. and J. Aggarwal. Spatio-temporal depth cuboid similarity Feature for activity recognition using depth camera. In *CVPR*,2012.

[6] P. Dollar., V. Rabaud., G. Cottrel. and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *PETS*,65-72,2005.

[7] A. Klaser., M. Marszelak. and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *BMVC*,2010.

[8] E. Ohn-Bar. and M.M. Trivedi. Joint angles similarities and HOG2 for action recognition. In *CVPRW*, 465-470, 2013.

[9] O. Oreifej. and Z. Liu. Histogram of oriented 4D normals for activity recognition from depth sequences. In *CVPR*, 2013.

[10] X. Yang. and Y. Tian. Super normal vector for activity recognition using depth sequences. In *CVPR*, 804-811, 2014.

[11] L. Xia., C.C. Chan and J.K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPRW*, 20-27, 2012.

[12] W. Li, Z. Zhang and Z. Liu. Action recognition based on a bag of 3d points. In *CVPRW*, 9-14, 2010.

[13] X. Yang and Y. Tian. Eigen-joints based action recognition using naive-Bayes-Nearest-Neighbor. In *CVPRW*,14-19, 2012.

[14] R. Vemulapalli, F. Arrate. and R. Chellappa. Human action recognition by representing 3D skeletons as points in a Lie group In *CVPR*, 588-595, 2014.

[15] M. Zanfir., M. Leordeanu. and C. Sminchisescu. The moving pose : An efficient 3d kinematics descriptor for low-latency action recognition and detection In *ICCV*, 2752-2759, 2013.

[16] C. Ellis, S.Z. Masood, M.F. Tappen, J.J. Laviola Jr, R. Sukthanka. Exploring the trade-off between accuracy and observational latency in action recognition. In *IJCV*, 101(3), 420-436, 2013

[17] G.T. Papadopoulos, A. Axenopoulos, P. Daras. Real-time skeleton-tracking-based human action recognition using kinect data In *MultiMedia Modeling*, 473-483, 2014

[18] C. Chang. and C. Lin. Libsvm : a library for support vector machines In *TIST*, 2(3) :27, 2011.

[19] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time Human Pose Recognition in Parts From a Single Depth Image. In *CVPR* , 2011

[20] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, In *CVPR*, 2005.

[21] J.R. Padilla-López, A.A. Chaaraoui, F. Flórez-Revuelta. A discussion on the validation tests employed to compare human action recognition methods using the MSR Action3D dataset, arXiv preprint arXiv :1407.7390,2014.

[22] M. Hammouche, E. Ghorbel, A. Fleury, S. Ambellouis Toward a real time view-invariant 3D action recognition In *VISAPP*, 2016.

[23] E. Ghorbel, R. Boutteau, J. Boonaert, X. Savatier, S. Lecoeuche 3D real-time human action recognition using a spline interpolation approach In *IPTA*, 2015.