MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU

FACULTE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT AUTOMATIQUE

MEMOIRE DE MAGISTER

en ***Automatique***

Option **Traitement d'Images et Reconnaissance de Formes**

Présenté par

HEMDANE Mohamed
Ingénieur U.M.B.B

Détection de personnes à partir d'images 3D et identification de leurs postures et de leurs mouvements par la caméra 3D Kinect.

Mémoire soutenu le: 22/10/2017 devant le jury d'examen composé de :

HAMMOUCHE Kamal	Professeur à l'UMMTO	Président
DIAF Moussa	Professeur à l'UMMTO	Rapporteur
LARABI Slimane	Professeur à l'USTHB	Examinateur
MAZOUZI Zohra	Professeur à l'UMMTO	Examinateur

Année 2017

Avant-propos

Ce mémoire a été effectué au laboratoire VISION ARTIFICIELLE ET AUTOMATIQUE DES SYSTEMES (LVAAS)

Mes vifs remerciements vont tout d'abord à Monsieur **DIAF Moussa**, Professeur à l'UMMTO pour m'avoir proposé le thème de ce mémoire et m'avoir dirigé, aidé et conseillé tout le long de notre travail.

Nous exprimons notre reconnaissance à Monsieur **HAMMOUCHE Kamal**, Professeur à l'UMMTO, pour m'avoir fait honneur en acceptant de présider le jury de ce mémoire.

Nos remerciements vont également à Monsieur **LARABI Slimane**, Professeur à l'USTHB pour avoir bien voulu faire partie du jury d'examen de notre mémoire.

Que Madame **AMEUR Zohra**, Professeur à l'UMMTO, trouve ici nos remerciements pour avoir accepté de faire partie du jury d'examen de notre mémoire.

Un grand merci à **Takieddine** de m'avoir aidé pendant toutes les étapes de ce travail, ainsi qu'à tous les gens qui ont aidé aux tests spécialement **Djameleddine**.

Je remercie aussi ma famille et tous mes amis.

Résumé

La détection de personnes et l'identification de leurs postures et leurs mouvements est un sujet qui peut intervenir dans différents applications allant des interactions gestuelles aux jeux vidéo, en passant par le suivi d'activités à domicile, la surveillance,…etc.

Pour qu'elles puissent être performantes et attractives, ces applications nécessitent la mise en œuvre d'outils de reconnaissance et d'interprétation des gestes humains, par des méthodes efficaces, rapides et ouvertes.

Dans ce mémoire et dans un premier volet, nous nous sommes orientés vers la détection de personnes à partir des images de profondeur de la Kinect. Un algorithme de détection a été développé, appelé histogramme de profondeurs orientées (HOD) inspiré de la méthode d'histogramme des gradients orientés (HOG) et des caractéristiques particulières du capteur Kinect. Nous effectuons une recherche multi-échelle informée de HOD basée sur une régression échelle profondeur et une utilisation d'images intégrales.

Dans un deuxième volet, une méthode pour la reconnaissance de poses en temps réel à partir d'un flux de squelette bruité tels que ceux extraits du capteur de profondeur Kinect, est présentée. Chaque pose est décrite en utilisant une représentation angulaire des articulations du squelette. Ces descripteurs servent à identifier des poses clés à travers un classifieur SVM multi-classes. Par la suite les poses clés seront utilisées pour reconnaitre les gestes à travers une forêt de décision qui permet la recherche efficace pour n'importe quelle séquence de poses clés qui composent un geste.

Le descripteur HOD a été testé sur une base d'images de profondeur et a donné des résultats satisfaisants, atteignant un taux de précision de 85%.

La machine d'apprentissage de poses clés était capable de reconnaître les poses clés des utilisateurs dans la plupart des cas, atteignant un taux de reconnaissance moyenne de 90,8%.

La forêt de décision a pu identifier efficacement les gestes en temps réel, des résultats excellents ont été obtenus dans la majorité des gestes.

Sommaire

Liste des figures

Liste des tableaux

Introduction générale

Si pour un humain, il est facile de distinguer son semblable autour de lui ou sur des images fixes ou dynamiques, pour un système de vision artificielle, cette tâche reste très complexe, alors que, ces derniers temps, le besoin de l'utilisation de moyens technologiques et informatiques permettant l'acquisition d'informations concernant la présence ou l'absence de personnes dans un environnement donné se fait sentir de plus en plus comme dans les systèmes de transport intelligent, la robotique, la télésurveillance, la domotique intelligente, l'indexation d'images ou de vidéos etc. De plus, une fois la présence d'une ou de plusieurs personnes détectée, leurs localisations, avec précision, sont souvent demandées. L'identification de la posture d'une personne dans un état debout ou en marche est aussi très recherchée particulièrement chez les piétons dans le cadre de la vidéosurveillance et pour les systèmes de vision embarqués dans des véhicules. Cependant, la détection de personne qui n'est étudiée qu'à partir de la fin des années 1990, présente encore beaucoup de complexité en raison de la variabilité des apparences des personnes liées aux articulations du corps humain et différents phénomènes d'occlusions.

Plusieurs méthodes de détection de personnes commencent de plus en plus à voir le jour comme celles basées sur les histogrammes de gradient orienté et celles basées sur des modèles statistiques par apprentissage supervisé, à partir de caractéristiques de forme ou d'apparence.

Par ailleurs, les nouvelles modalités d'interaction basées sur la vidéo ont suscité de nouveaux besoins auprès des utilisateurs. C'est ainsi que des périphériques comme *l'EyeToy* de Sony ou la *Kinect* de Microsoft ont offerts aux utilisateurs la possibilité d'utiliser leurs mains, leurs corps et leurs mouvements pour interagir avec les séquences d'images mises en scène par un programmé implémenté dans un ordinateur. Ces nouveaux périphériques nécessitent de nouvelles approches pour l'interprétation de ces mouvements et leur traduction en ordres et ce en vue de toucher un public large, ce qui exige plus de rapidité et plus de souplesse avec une possibilité, à

l'utilisateur, d'ajouter de nouveaux ordres gestuels. En effet, l'interprétation automatique des gestes et des actions existe depuis que les caméras numériques sont disponibles et parmi les premiers travaux dans ce domaine, on peut notamment citer ceux relatifs à la marche, l'analyse de mouvement de foule etc. La précision, la souplesse et la rapidité sont souvent les trois contraintes à remplir dans ces types d'applications où, parfois, un compromis entre ces différentes exigences s'impose. Pour certaines applications médicales, par exemple, nous aurons tendance à privilégier la précision, alors que dans une application ludique nous chercherons en priorité la rapidité et l'adaptabilité. Chaque solution doit trouver son propre équilibre entre ces contraintes en fonction de son contexte d'application. En plus de ces contraintes, d'autres problèmes peuvent surgir comme l'extraction des informations de postures de façon fiable et constante au cours du temps alors qu'elles peuvent être noyées dans du bruit. Comme autre problème, nous avons aussi à citer les grandes variations de style dans une action de reproduction d'un même geste. Une action peut être une composition de mouvements tels le «*geste*», l'«*action*», l'«*interaction* » et l' «*activité de groupe*».

Un « *geste* » est défini comme le mouvement élémentaire des parties du corps d'une personne. Il s'agit de la décomposition sémantique la plus élémentaire, c'est- à-dire, la plus proche du mouvement perçu, comme «agiter le bras», «lever une jambe», etc. Une « *action* » est l'activité d'une seule personne qui peut être composée de plusieurs gestes, arrangés temporellement, comme «marcher», «sauter», etc. Une «interaction» est une activité humaine qui implique, soit, deux humains au moins agissant ensemble, éventuellement avec un ou des objets, soit un seul humain mais agissant avec au moins un objet.

Dans le cadre de ce mémoire structuré en trois chapitres, nous nous intéressons à la reconnaissance automatique de gestes ou mouvements élémentaires des parties du corps d'une personne.

Dans le premier chapitre, nous présentons un contexte général et un état de l'art des méthodes et des outils fondamentaux pour la détection de

personnes et l'identification de leurs gestes qui incluent l'extraction des vecteurs caractéristiques et les différents algorithmes de classification.

Dans le deuxième chapitre, nous présentons un algorithme de détection de personnes en temps réel à partir des images de profondeur de la Kinect. Nous présentons d'abords le descripteur nommé histogramme des profondeurs orientées puis une classification basée sur les machines à vecteurs de support.

Dans le troisième chapitre, nous présentons une méthode pour la reconnaissance des poses et des gestes en temps réel à partir d'un flux de squelette bruité. Chaque pose est décrite en utilisant une représentation angulaire des articulations du squelette. Ces descripteurs servent à identifier des poses clés à travers un classifieur SVM multi-classe. Par la suite les poses clés seront utilisées pour reconnaitre les gestes à travers une forêt de décision.

Ce mémoire se termine par une conclusion générale et des perspectives ainsi que des références bibliographiques.

Chapitre 1

Etat de L'art

1.1 Introduction

La détection de personnes et la compréhension automatique de l'activité humaine est un sujet qui a connu un progrès considérable cette dernière décennie surtout avec les évolutions technologiques telles les caméras à capteur de profondeur. Dans ce domaine, s'inscrit la reconnaissance automatique de gestes qui regroupe l'ensemble des techniques visant à identifier et traduire les mouvements et les gestes du corps humain pour différentes applications comme la vidéosurveillance et la communication homme-machine.

Techniquement, l'étude de la question dépend du dispositif de capture utilisé comme la caméra 2D, le capteur de profondeur et autres. La séparation de l'objet d'intérêt de son environnement ainsi les techniques de description et d'interprétation jouent un rôle important dans l'efficacité de la procédure.

Ainsi, différentes méthodes utilisant les images de profondeur ont été proposées dans la littérature. Ces méthodes peuvent être divisées en deux catégories différentes à savoir les méthodes basées correspondance et les méthodes statistiques. Les approches de la deuxième catégorie, dans

lesquelles le choix de descripteurs appropriés est essentiel ont donné des résultats intéressants.

Dans ce chapitre, il s'agit de présenter, de manière succincte, la détection de personnes et l'identification de leurs mouvements à partir des images de profondeur ainsi qu'un état de l'art des approches utilisées.

1.2 Détection de personnes et reconnaissance de gestes

Notons d'abord que le "geste" est difficile à définir de manière précise. Dans [1], il est mentionné que « *tout le monde prétend savoir ce qu'est un geste, mais personne ne peut vous l'expliquer précisément*». Selon le dictionnaire Larousse [2], c'est un " *Mouvement du corps, principalement de la main, des bras, de la tête, porteur ou non de signification* ". Dans [3], le geste est défini comme un «*Morceau d'une trajectoire spatio-temporelle qui possède une trajectoire stéréotypée autorisant une grande variabilité*».

De façon générale, un geste peut être défini par une approche *fonctionnelle* qui se réfère aux fonctions qu'il peut exécuter dans des situations spécifiques.

1.2.1 Typologie de gestes

La principale classification de gestes est proposée par Mcneill [4] et légèrement modifiée par Pavlovic [5] pour inclure les mouvements (Fig.1.1). Dans cette figure, les mouvements sont regroupés dans deux classes, les gestes qui sont des mouvements significatifs et les gesticulations qui sont des mouvements non significatifs. Les gestes eux mêmes sont regroupés dans deux différentes classes, les gestes manipulatifs et les gestes communicatifs ou sémiotiques. La première catégorie regroupe les actions matérielles sur l'environnement visant à le modifier comme déplacer un objet ou à le percevoir comme le toucher. La deuxième regroupe les gestes porteurs d'informations destinées à l'environnement. Ces gestes servent souvent à préciser le discours oral.

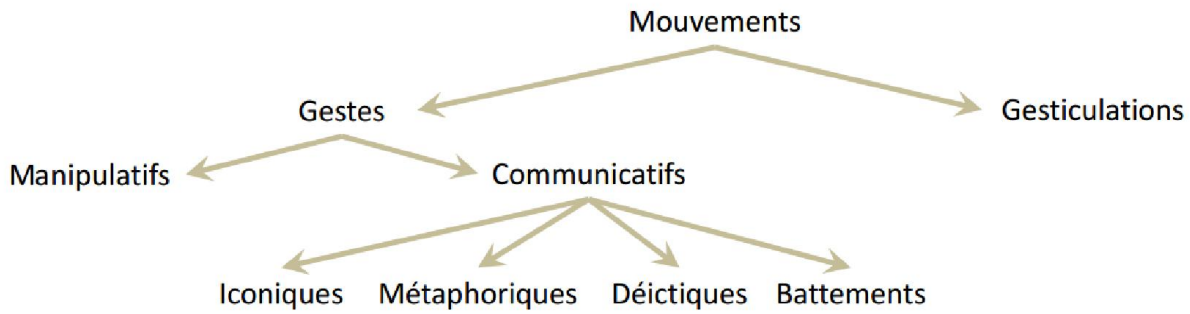

Fig.1.1 Taxonomie du geste [4] [5]

Les gestes communicatifs eux mêmes sont partagés en quatre groupes dits *iconiques, métaphoriques, déictiques* et *battements*. Les Ioniques véhiculent par leur forme et par leur mouvement le contenu relatif au contenu linguistique concomitant comme, par exemple, le geste de l'escalier en colimaçon. Les métaphoriques présentent les images des notions abstraites. Ils diffèrent des iconiques par l'impossibilité de présenter visuellement ce qu'ils véhiculent comme, par exemple le geste de la solidarité dans la jonction des deux mains. Les déictiques ont un caractère narratif et décrivent des traits caractéristiques d'un objet concret évoqué dans la conversation comme montagne, un cercle etc. Ces gestes portent une relation perceptuelle avec des entités concrètes et des événements. Quant aux battements, ils rythment le discours et sont souvent brefs et d'appuis. Ces gestes, de par leur régularité, fournissent une structure temporelle de communication et facilitent la recherche lexicale des mots.

On différencie également les gestes dynamiques des gestes statiques. Un *geste statique*, encore appelé posture, concerne la configuration du corps ou d'une partie du corps à un moment fixe dans le temps alors que le *geste dynamique* désigne une succession continue de postures.

Notons finalement que les gestes peuvent être distingués en fonction des parties du corps impliquées. On distingue généralement les gestes de la main et du bras et les gestes de la tête et du visage. Les gestes de la main et du bras forment la principale catégorie de gestes interactifs. La main permet de réaliser des gestes précis et complexes. Les recherches autour de ces gestes concernent principalement la reconnaissance de positions de la main, l'interprétation du langage des signes et le développement d'interface

homme-machine permettant la manipulation et l'interaction avec des données ou des éléments d'un environnement virtuel. Dans les gestes de la tête et du visage, peu ont une signification spécifique alors que l'orientation de la tête est très utile pour la détection du champ de vision. Quant aux gestes impliquant tout le corps, les recherches dans ce domaine s'intéressent à tout le corps en interaction avec son environnement comme l'analyse des gestes d'un athlète pour améliorer ses performances.

1.2.2 Capture de mouvements

On distingue plusieurs types de systèmes de capture de mouvements. Les systèmes mécaniques, les systèmes acoustiques, les systèmes optoélectroniques et les systèmes magnétiques. Les systèmes mécaniques sont encombrants, surtout lorsqu'il s'agit de capter le corps entier. Les systèmes optoélectroniques sont précis (erreur de position 3D inférieure à 1mm) mais l'inconvénient est qu'ils sont sensibles aux conditions lumineuses. Les capteurs magnétiques sont sensibles aux sources de bruit, comme la présence d'éléments ferromagnétiques.

Certains dispositifs comme la Kinect nous offrent la possibilité de détecter des personnes et d'avoir les articulations du squelette en 3D comme cela est proposé dans l'algorithme de détection proposé par Shotton [6]. Cet algorithme basé sur les forêts d'arbres aléatoires permet d'avoir 20 articulations de la personne à une fréquence de 30 trames par seconde.

1.2.3 Reconnaissance de gestes

La majorité des algorithmes de reconnaissance de gestes utilisent une approche statistique basée sur des descripteurs et une classification. Les descripteurs doivent faciliter la tâche de la classification. Ils doivent être discriminants pour présenter les informations pertinentes au problème posé. Mathématiquement, ils doivent être en mesure de maximiser les distances interclasses et minimiser les distances intra-classe. Ils doivent aussi être robustes aux bruits et aux variations d'échelles et de dimension réduite par l'application de l'analyse en composantes principales qui permet de réduire la dimensionnalité du vecteur de caractéristiques et garder les informations pertinentes par la construction d'une autre base.

Quant à la classification, sa tâche a pour objectif soit de déterminer si un geste appartient ou non à une classe de gestes, soit de déterminer la classe la plus probable parmi un ensemble prédéfini de classes. Plusieurs algorithmes de classification sont utilisés dans le domaine de la reconnaissance de gestes parmi lesquelles, on peut citer :

Les k-moyennes : L'algorithme des k-moyennes est l'un des plus simples algorithmes d'apprentissage non supervisé, appelé aussi algorithme des centres mobiles ou K-means [7].

Les k plus proches voisins : La méthode des k plus proches voisins consiste à déterminer pour chaque nouvel objet non classé, la liste des plus proches voisins parmi les objets déjà classés. L'objet est affecté à la classe qui contient le plus d'individus parmi ces plus proches voisins. Cette méthode utilise souvent la distance euclidienne. L'algorithme nécessite le calcul de la distance avec tous les exemples qui rend coûteux en termes de temps de calcul, mais avec l'utilisation des heuristiques on peut réduire le temps de prédiction [8].

Les SVM : Les Machines à Vecteurs de Support (SVM) ont été introduites par Vapnik [9] et ont été fréquemment exploités en classification principalement la reconnaissance de gestes. Basés sur deux concepts à savoir, les fonctions noyaux permettant de changer d'espace de représentation et la notion de marge maximale pour effectuer une séparation linéaire. Initialement le problème est celui de la recherche d'une séparation linéaire entre deux classes. Vapnik a démontré que la solution optimale est celle qui maximise la marge, c'est-à-dire, la distance entre la frontière et les objets les plus proches de cette dernière (Fig.1.2). Il n'est pas toujours possible de trouver une solution pour une séparation linéaire. En pratique, un paramètre C est ajouté pour pénaliser chaque objet situé du mauvais coté. Une fonction noyau peut être utilisée pour transformer l'espace initial en un espace de dimension supérieure dans lequel une séparation linéaire est possible. Les principaux noyaux utilisés sont:

– Le noyau linéaire : $K(x,y) = x^t \cdot y$ (produit scalaire).

– Le noyau polynomial : $K(x,y) = (x^t \cdot y + c)^d$

– Le noyau gaussien (ou RBF) : $K(x,y) = exp(||x-y||/2\sigma^2$

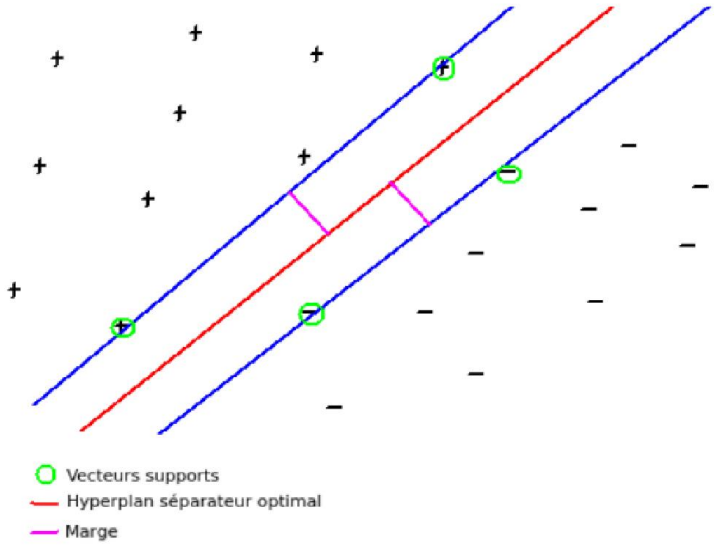

Fig.1.2 Exemple de séparation linéaire

Réseaux de neurones artificiels : Rappelons que les réseaux de neurones artificiels sont des structures de neurones dont les entrées sont les sorties d'autres neurones, les connexions entre neurones sont associées à des poids. Différentes structures sont possibles, la principale est celle appelée perceptrons multicouches qui sont des réseaux de neurones organisés en couches superposées [10]. Ils contiennent une couche en entrée, une autre en sortie et un ensemble de couches entre les deux. Chaque neurone reçoit un stimulus des neurones de la couche qui le précède et transmet un stimulus aux neurones de la couche qui vient après. On associe à chaque liaison entre deux neurones un poids. L'apprentissage se fait par rétro-propagation, les poids sont modifiés pour avoir le comportement voulu.

Arbres de décisions : Un arbre de décision est une structure contenant des arbres est des feuilles, chaque nœud étant associé à un critère de décision pour permettre d'aller à droite ou à gauche à un nœud fils [11]. Les feuilles de l'arbre correspondent à un critère d'arrêt. Chaque feuille correspond à une décision ou la classe affectée au descripteur en entrée. L'apprentissage consiste à choisir les tests associés à chaque nœud. Ce choix est la clé pour avoir un arbre discriminant.

Forêts aléatoires : Ils sont basés sur les arbres de décisions. L'idée est de générer aléatoirement un ensemble d'arbres en déterminant aléatoirement le test effectué à chaque nœud ou en sélectionnant pour chaque arbre un

sous-ensemble de variables explicatives à exploiter [12]. Il est possible en phase d'entrainement de sélectionner un sous-ensemble de ces arbres générés aléatoirement. La décision finale se fait par un vote de toutes les décisions individuelles prises par chacun des arbres. Le principal avantage des forêts aléatoires est que sont naturellement faites pour une classification multi-classes.

1.3 Etat de l'art

Dans cette section, nous allons présenter quelques méthodes utilisées pour la détection de personnes et l'identification de mouvements à partir des images de profondeur.

1.3.1 Détection de personnes à partir des images de profondeur

Récemment, quelques méthodes de détection de personnes à partir des images de profondeurs ont été proposées. En 2011, Spinello [13] a proposé une méthode de détection de personnes appelé Histogramme des Profondeurs Orientées (HOD). L'approche consiste en deux opérations. La première permet la détection de personnes à partir des images de profondeur sur la base des histogrammes (HOD) inspirée de la méthode de l'Histogramme des Gradients Orientés (HOG). La deuxième est la recherche multi-échelle informée de HOD basée sur une régression échelle de profondeur et une utilisation d'images intégrales [14]. La méthode considère une fenêtre de détection de taille fixe subdivisée en une forte densité de réseau uniforme de cellules. Pour chaque cellule, les orientations de gradient de profondeurs sont calculées et recueillies dans un histogramme 1D. L'intuition est que l'aspect local et la forme peuvent être caractérisés par une distribution de gradients locaux sans la connaissance précise de leur position dans la cellule. Les groupes de cellules adjacentes, appelées blocs, sont utilisés pour la normalisation. Le descripteur, construit par la concaténation de tous les histogrammes de blocs est alors pris pour entraîner un SVM linéaire. Pour la détection de personnes, la fenêtre de détection est glissée sur l'image à plusieurs échelles. Pour chaque

position et échelle, les descripteurs HOD sont calculés et classés avec le SVM entraîné. Avec l'utilisation des images intégrales [14], Spinello propose une solution beaucoup plus rapide capable de tester l'échelle en O(1). Les images Intégrales sont une technique efficace pour calculer la somme des valeurs dans une zone rectangulaire d'une grille. La valeur de chaque point d'image est la somme de toutes les valeurs ci-dessus et à la gauche du point.

La construction de l'image intégrale elle-même est d'une procédure O (N), où N dépend de la taille de l'image d'origine. Le principal avantage d'images intégrales est le calcul d'une zone intégrale avec seulement 4 soustractions. L'auteur étend le concept en tenseurs intégrales, images intégrales multicouches avec autant de couches que les échelles *S* se trouvant dans l'image. Chaque couche, dans le tenseur intégral, est une image binaire dont les pixels non blancs correspondent à l'échelle de la couche. Ceci permet de tester de manière très efficace si une fenêtre de recherche donnée contient au moins un pixel d'une échelle particulière. La construction du tenseur intégral doit être faite qu'une seule fois par image.

Dans la phase de détection, une échelle *s* de *S* est sélectionnée. Ensuite, pour chaque position de la fenêtre de recherche, le test est pris comme une zone intégrale sur la fenêtre de recherche dans la couche du tenseur intégral qui correspond à *s*. Si le résultat est supérieur à zéro, il existe au moins un pixel de profondeur compatible sous la fenêtre et un descripteur HOD est calculé. Sinon, la fenêtre de détection n'est pas considérée et le processus se poursuit.

En 2013, Yujie [15] propose une approche de détection de personnes à partir des images de profondeur de la Kinect. D'abord, un filtrage est appliqué pour réparer les défauts de la carte de profondeur. Ensuite, une base d'image de profondeur contenant des personnes dans différentes postures est construite. Finalement, en introduisant les masques de Kirsch [16] et un code à trois valeurs à une LBP [17] modifiée conduit à un descripteur appelé motifs locaux de direction ternaire. Yujie utilise une fenêtre de détection de taille 64×128 pixels subdivisée à des blocs rectangulaires qui ne chevauchent pas. Les histogrammes des motifs locaux calculés à partir de

chaque bloc sont accumulés pour avoir un descripteur de dimension 3776. Pour la tâche de classification un SVM est utilisé. Le principal inconvénient est la sensibilité aux bruits du descripteur puisqu'il utilise une LBP.

En 2010, Ikemura [18] propose une détection de personnes à partir des images de profondeurs obtenues via une caméra à temps de vol qui peut fonctionner dans des scènes complexes. L'approche proposée calcule des caractéristiques de similarité en profondeurs de deux régions locales, la fenêtre de détection est subdivisée à des régions locales qui sont des cellules de 8×8 pixels, deux cellules sont choisies. Ikemura calcule des histogrammes de profondeurs à partir de l'information de profondeur de chaque cellule, après une normalisation est appliquée à chaque histogramme. Dans cette méthode, la taille de la fenêtre de détection est 64×128 pixels donc elle contient 8×16 cellules. Il y a 492 régions rectangulaires en faisant varier la taille de la cellule unité de 1×1 jusqu'à 8×8. Le vecteur de caractéristiques final est de dimension 120 786. Pour réduire le temps de calcul, les histogrammes intégraux sont utilisés pour calculer les histogrammes de profondeurs. La classification est faite par un AdaBoost [19]. La méthode a achevé un taux de détection de 95% à 10 trames par seconde. Cette méthode présente l'intérêt qu'elle propose une solution pour le problème de l'occlusion. Le défaut majeur de cette approche est la dimension du vecteur de caractéristiques.

En 2011, Lu [20] et al ont proposé une méthode de détection de personnes basée sur la carte de profondeur en utilisant un modèle à deux étages contenant un modèle 2D de la tête et un modèle 3D de surface de la tête.

Etant donnée une image de profondeur, d'abords le bruit est réduit et l'image est lissée en utilisant un filtre médian. Par la suite, ils utilisent le détecteur de contour de Canny [21] et un modèle 2D de la tête pour accélérer la recherche qui permet d'avoir le minimum de faux négatifs et qui peut avoir un taux élevé de faux positifs qui seront réduits dans l'étape qui suit. Le modèle 3D de la tête est utilisé pour examiner toutes les régions détectées comme positives dans l'étape précédente. Un hémisphère est pris comme un modèle 3D et comparé avec la région, l'erreur quadratique est calculée entre le modèle et la région détectée, un seuil est accepté pour décider s'il s'agit

d'une tête ou non. La dernière étape consiste à développer un algorithme d'extension pour avoir toutes les parties du corps de la personne, il est assumé que la profondeur de la surface de la silhouette est continue et sa variation est petite. La règle d'extension est basée sur la similarité en profondeur entre la région et les pixels voisins.

Le principal inconvénient de cette approche est qu'elle débute par la détection de la tête, si la tête n'est pas détectée alors la personne n'est pas détectée.

1.3.2 Identification de mouvements

Un ensemble de méthodes d'identification de mouvements est présenté dans cette partie. Ces méthodes diffèrent principalement dans l'extraction des descripteurs et la classification.

En 2013, Ofli et al. [22] proposent un descripteur basé sur la sélection d'articulations contenant le plus d'information. Ils utilisent des séries temporelles de positions d'articulations 3D segmentées, sur chaque segment, la quantité d'information associée à chaque articulation est calculée, les séries sont ensuite arrangées par ordre de quantité d'information décroissante.

Pour sa part Hussein et al. [23] proposent, en 2013 aussi, un descripteur basé sur des matrices de covariance. Ils utilisent les positions d'articulations 3D évoluant dans le temps. Le descripteur proposé est la matrice de covariance de l'ensemble des coordonnées de toutes les articulations. L'aspect temporel du mouvement est traité en utilisant une hiérarchie inspirée des travaux de Lazebnik et al. [24]. Pour la tâche de classification, Hussein et al utilisent un SVM à noyau linéaire.

Wanqing Li et al. [25] ont présenté en 2010 une méthode basée sur un graphe d'action. Ce travail est un étendu d'une autre publication de 2008 [26] similaire mais exploitée sur des données 2D. Les données sont obtenues à partir d'une caméra à capteurs de profondeur. Les contours de la silhouette sont obtenus à partir des projections sur chacun des plans (x, y), (x, z) et (y, z), puis un certain nombre de positions sont échantillonnées sur chacun des trois contours afin de réduire la complexité. Ce descripteur est utilisé pour évaluer par un modèle de mélange gaussien les postures qui

représentent les états d'un graphe d'action. Un modèle de Markov à états visibles est exploité pour la classification.

En 2012, Wang et al [27] proposent plusieurs méthodes. Ils utilisent les articulations 3D du squelette. Pour chaque articulation un vecteur évoluant au cours du temps composé de sa position relative par rapport à chacune des autres articulations ainsi une information concernant l'occupation de l'espace autour de l'articulation. Les positions relatives sont utilisées pour avoir l'information concernant la configuration du squelette, l'occupation de l'espace autour de l'articulation est utilisée pour conserver l'information relative à l'interaction entre le sujet et les objets qui l'entourent.

Par la suite, une hiérarchie temporelle inspirée des travaux de Lazebnik et al. [24] est utilisée et sur chacun des segments temporels, les principaux coefficients de la transformée de Fourier sont calculés. La segmentation conserve l'aspect temporel tandis que la transformation de Fourier permet d'obtenir une résistante au bruit et au décalage temporel entre deux séries. La concaténation des vecteurs est le descripteur final.

Pour la classification, les auteurs sélectionnent un sous ensemble d'articulations discriminant nommé *actionlets*. Tout d'abords le pouvoir discriminant de chaque articulation est évalué en entraînant un SVM pour chaque articulation, ils déterminent ensuite l'ensemble des *actionlets* respectant un certain seuil de confiance. Sur chacun de ces *actionlets* un nouveau SVM est entrainé. Les *actionlets* sélectionnés servent à entrainer un dernier SVM dont le noyau constitué d'une combinaison convexe des noyaux exploités auparavant.

Oreifej [28] a présenté en 2013 un histogramme des normales en 4D comme descripteur. Les données sont obtenues via une caméra à capteur de profondeur Kinect. Les normales sont calculées dans un espace à quatre dimensions (les trois dimensions spatiales et la dimension temporelle). Un ensemble de projecteurs sont définis et qui représentent les 120 sommets d'un *polychoron* régulier à 600 cellules. Le *polychoron* représente l'extension d'un polygone dans un espace à quatre dimensions. Finalement, un vecteur de dimension 120 est obtenu en sommant pour chacun des 120 projecteurs

les projections de chaque normale avec lui-même. Un SVM est utilisé pour la classification.

Sylvain et al [29], ont exploité des modèles markoviens à états cachés pour apprendre des classes de mouvements. Ils utilisent des coordonnées ou des rotations d'articulations en 3D. Des extrema locaux sont extraits de chaque série temporelle. Pour la phase d'entrainement, à chaque extremum est associé un état caché, une fonction probabiliste des observations est associée à chaque état.

Dans [30], Clément compare différents modèles de Markov cachés.

Mitra cite autres études [31] [32] [33] [34] dans leur revue [35] publiée en 2007 exploitant les chaines de Markov.

Bashir et al. [36] utilisent des modèles markoviens à états cachés sur des séries temporelles segmentées et réduites avec une ACP. Ils utilisent les positions (x, y) d'objets extraits d'une vidéo. Les séries temporelles sont segmentées en fonction de discontinuités dans la trajectoire. Ils associent à chaque segment un ensemble de coordonnées(x, y). Tous les segments sont accumulés en une matrice. Une ACP [37] est appliquée sur cette matrice. Sur ce sous-espace obtenu par la ACP, un algorithme de classification spectrale combiné avec les k-moyennes est utilisé.

En 2010, Bevilacqua et al. [38] ont proposé un modèle de suivi de mouvement en temps réel qui s'adaptent à n'importe quel type de données multidimensionnelles échantillonnées régulièrement. La méthode s'applique à de longues séries temporelles pour lesquelles le nombre d'états cachés serait trop important pour effectuer des calculs en temps réel. La méthode calcule en continu l'état le plus vraisemblable et utilise une fenêtre glissante autour de cet état pour limiter le nombre d'états possibles.

Une autre approche utilisée fréquemment est la déformation temporelle dynamique, un algorithme qui permet la comparaison des séries temporelles. Elle établit une mesure de similarité entre des paires de séries en prenant en compte les variations de rythme au cours du temps. L'exploitation de la déformation temporelle en reconnaissance de mouvements a fait l'objet de plusieurs publications [39] [40] [41]. En général, la classification est faite par le k plus proches voisins.

En 2012, Miranda et al [42] proposent une méthode pour la reconnaissance des poses en temps réel à partir d'un flux de squelette bruité, tels que ceux extraits du capteur de profondeur Kinect. Chaque pose est décrite en utilisant une représentation angulaire des articulations du squelette. Ces descripteurs servent à identifier des poses clés à travers un classifieur SVM multi-classe. Par la suite les poses clés sont utilisées pour reconnaitre les gestes à travers une forêt de décision qui tient en compte les contraintes de temps/vitesse. Ils convertissent l'ensemble des points représentant les nœuds du squelette à une représentation angulaire réduite (de dimension 18). La représentation est un étendu de celle de Raptis et al. [43] pour une représentation angulaire pure. Elle fournit une invariance à l'orientation du capteur et réduit la redondance et la dimensionnalité, tout en préservant les informations pertinentes pour la classification des poses clés. Après l'entraînement des poses clés, ils étiquètent des exemples de gestes. Les poses clés apparaissant dans chaque geste sont automatiquement identifiées à partir des machines SVM. Après l'entraînement, une forêt de décision est optimisée pour permettre la recherche efficace pour n'importe quelle séquence de poses clés qui composent un geste. Pour chaque performance d'un geste, les poses clés sont accumulées dans une mémoire tampon circulaire en vérifiant dans la forêt de décision si la séquence complète un geste connu. Cela évite la nécessité d'une pose initiale/neutre. Les nœuds de la forêt peuvent éventuellement considérer les contraintes temps/vitesse. Même lorsque différents utilisateurs effectuent le même geste avec une durée différente entre les poses clés, la forêt de décision apporte une solution efficace et robuste à ce problème d'alignement temporel.

1.4 Conclusion

Dans ce chapitre, nous avons présenté les approches les plus utilisées pour les problèmes de la détection de personnes et l'identification de mouvements à partir des images de profondeurs.

Puisque l'utilisation des caméras de profondeurs est très récente, peu de travaux ont étés proposés pour la détection de personnes mais plus de travaux d'identification de mouvements ont été proposés utilisant le flux de squelette, principalement de la Kinect.

Les problèmes rencontrés sont variés. Ils se résument en l'extraction de la personne de son environnement, l'extraction de ce qui caractérise le geste (le descripteur), la tâche de classification et, enfin, fournir une interaction en temps-réel.

Cependant, il a été constaté que les méthodes basées correspondance sont moins robustes que celles basées sur les vecteurs de caractéristiques qui, avec l'utilisation, d'un bon classifieur, conduisent à des résultats meilleurs.

Dans ce travail, nous avons opté pour les méthodes statistiques et la classification automatique par la méthode des SVM dans lesquelles le choix de descripteurs appropriés est essentiel.

Chapitre 2

Détection de personnes

2.1 Introduction

De systèmes intelligents de surveillance vidéo basés sur la détection et la compréhension automatique de l'activité humaine suscitent de plus en plus d'intérêt. Pour la détection de personnes, différentes méthodes ayant atteint une grande précision ont été proposées. Ces méthodes peuvent être regroupées en deux catégories différentes. Dans la première, on trouve les méthodes basées correspondance et dans la deuxième, les méthodes statistiques. Les approches de la deuxième catégorie, dans laquelle le choix des descripteurs est important, sont robustes et fournissent des résultats intéressants. Même si, au cours des dernières années, plusieurs descripteurs pour les images visibles RGB ont été proposés, l'histogramme des gradients orientés HOG donne de meilleurs résultats [44]. La carte de

profondeur représentant l'information spatiale d'un objet, ses applications font objet de plus en plus de travaux de recherche au cours des dernières années. Ainsi, Yujie [15] utilise les motifs locaux de direction ternaire (MLDT), un descripteur pour la détection de personnes avec SVM comme classifieur. Ikemura [18] a proposé une méthode de détection à base de fenêtres en utilisant la caractéristique de similarité relationnelle en profondeur. Lu [20] a proposé une méthode de détection de personnes basée sur la carte de profondeur en utilisant un modèle à deux étages contenant un modèle 2D de la tête et un modèle 3D de surface de la tête. Spinello [13] a proposé une méthode de détection de personnes HOD, une approche basée sur l'approche de HOG. Vu l'efficacité du descripteur HOG, l'autre descripteur HOD (Histogrammes des Profondeurs Orientées) a été proposé. Dans ce chapitre, nous présentons les différentes étapes pour construire le descripteur HOD et son utilisation qui repose sur une classification par un SVM. Ainsi, nous développons une détection de personnes à partir d'images de profondeur HOD. Nous effectuons une recherche multi-échelle basée sur une régression échelle profondeur et une utilisation d'images intégrales [14].
Le capteur Kinect étant le moyen d'acquisition des images, la présentation, ci-après, de ses caractéristiques nous semble nécessaire.

2.2 Le capteur Kinect

Notons d'abord que le capteur Kinect est un périphérique de Microsoft destiné à la console de jeux vidéo Xbox 360, qui permet de contrôler des jeux vidéo sans utiliser de manettes. Il est basé sur une technologie logicielle développée et sur une caméra spécifique qui interprète les informations sur la scène 3D obtenue à travers une lumière infrarouge structurée et projetée en continu. Ce système de scanner 3D appelé «*Light Coding*» utilise une variante de la reconstruction 3D. Le dispositif comporte une caméra RGB, un capteur de profondeur et un microphone multi-réseau exécutant un logiciel propriétaire de capture des mouvements du corps en 3D, de reconnaissance faciale et vocale. Le capteur de profondeur se compose d'un projecteur laser infrarouge combiné à un capteur CMOS monochrome qui capture des données vidéo en 3D dans toutes les conditions de lumière

ambiante. Le rayon de détection de ce capteur est réglable, et le logiciel de Kinect [45] est capable de calibrer automatiquement le capteur en fonction de l'environnement physique pouvant accueillir la présence des meubles ou d'autres obstacles. Une caméra RGB, comme toute *webcam*, fournit une image représentant la lumière réfléchie par les éléments dans la scène. Cependant, le principe physique de capture de la Kinect est différent car il s'agit d'une caméra vidéo 3D qui fournit des images de la profondeur de la scène, à savoir la distance entre la caméra et les objets présents dans la scène. L'image de profondeur se présente en niveaux de gris et pour la capturer, il existe un grand nombre de techniques. Certaines se basent sur des capteurs spécifiques permettant de réaliser une mesure physique de la profondeur. D'autres techniques tentent de calculer la profondeur de manière indirecte en se basant sur des images couleur, par exemple, en examinant les ombres ou le mouvement dans une scène. L'utilisation de deux caméras pour la vision stéréoscopique doit sa popularité à l'analogie avec le système visuel humain et la disponibilité de caméras couleur à bas prix. Cependant, les systèmes de vision stéréoscopiques ne sont capables que de calculer la profondeur de la scène pour un ensemble restreint de pixels correspondant à des zones de la scène ayant une forte structure locale, reconnaissable facilement d'une image à l'autre. Par exemple, il est impossible d'obtenir par stéréoscopie une information fiable sur la profondeur d'un objet de couleur unie comme, par exemple, une feuille blanche qui occuperait le champ entier de la caméra. Toutefois, il existe des caméras 3D permettent de mieux capturer la profondeur d'une scène. Ces caméras 3D, dites actives, possèdent leur propre source de lumière pour calculer la profondeur. Par exemple, les caméras dites à temps de vol envoient une lumière visible ou infrarouge et mesurent le temps entre l'émission d'une onde et la réception du signal réfléchi par la scène, à la manière d'un écho. Comme la vitesse de la lumière est une constante, la distance s'obtient en multipliant le temps mesuré par la vitesse de la lumière en espace libre. Le mode de fonctionnement de la Kinect est encore différent. En effet, la Kinect est une caméra 3D active de type stéréoscopique. Comme le montre la figure 2.1, elle comprend une source de

lumière infrarouge «structurée» et une caméra infrarouge. La méthode d'acquisition se base sur les mêmes principes géométriques que ceux utilisés pour la stéréoscopie. En stéréoscopie, on exploite le fait que plus un objet central est proche des deux caméras, plus il est décalé vers la droite dans l'image de gauche et vers la gauche dans l'image de droite. De même, pour un objet distant, le décalage est plus faible. Cependant, grâce à sa propre source lumineuse, la Kinect lève les indéterminations inhérentes à la stéréoscopie dans les zones dépourvues de texture.

Finalement, on peut résumer en disant que la Kinect dispose d'une caméra couleur, en plus du dispositif stéréoscopique de vision 3D comme le montre la figure 2.1. Pour chaque pixel de l'image, elle fournit en fin de compte 4 informations à savoir, les 3 composantes de couleur et la profondeur (sauf aux bords de l'image). D'un point de vue technique, la Kinect produit des images ayant une résolution de 640 × 480 pixels à une cadence de 30 images par seconde. L'angle d'ouverture de la caméra 3D de la Kinect est d'environ 58° [46] et est plus petit que celui de la caméra couleur. Ainsi, la zone de la scène vue par la caméra couleur englobe toujours celle vue par la caméra de profondeur. De plus, En effet, la lumière émise par la Kinect permet de donner une texture facilement identifiable à toute la scène filmée. La figure 2.2 montre un exemple de mire, d'allure aléatoire, projetée en infrarouge (non visible) par la Kinect.

Fig.2.1 Face avant de la Kinect

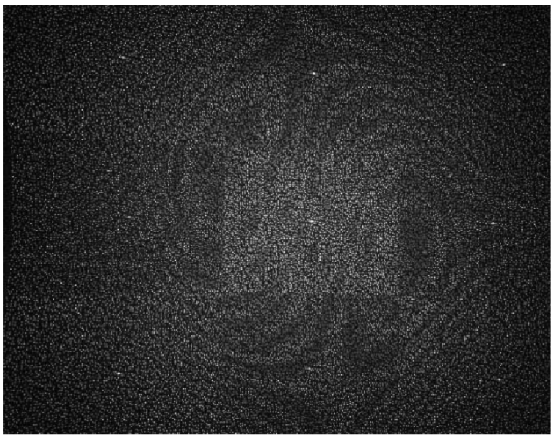

Fig.2.2 Mire émise par la Kinect perçue par une caméra infrarouge

Notons que l'image de profondeur a une résolution de pixel 640×480 à 11 bits par pixel. Fait intéressant, ce n'est pas tous les bits sont utilisés pour le codage de la profondeur: les valeurs hors de portée (par exemple ci-dessous gamme minimum) sont marqués avec la valeur de $V_{max} = 1084$ tandis que la gamme minimum est déterminée expérimentalement pour avoir $V_{min} = 290$. Ainsi, seulement 794 valeurs (10 bits) sont utilisées pour obtenir des informations de profondeur de codage dans chaque pixel. La relation entre la valeur de profondeur initiale v et la profondeur métrique en mètres d a été déterminée expérimentalement dans [13] et donnée par l'expression :

$$d = \frac{8 \, . B \, . F_x}{(V_{max} - v)} \tag{2.1}$$

où $B = 0.075m$ correspond à la distance entre le projecteur infrarouge et la caméra infrarouge (la base), et F_x, la distance focale de la caméra infrarouge dans la direction horizontale. Les valeurs négatives de d sont écartées. La fonction précédente est une relation hyperbolique similaire à la façon dont la profondeur est déterminée à partir des correspondances point-à-point dans les systèmes de caméras stéréo.

2.3 Détection de personnes dans les images de profondeur

Dans cette section, nous présentons le détecteur proposé. Nous donnons d'abord un résumé du détecteur HOG pour des données d'image RGB. Ensuite, nous introduisons le HOD, un autre détecteur pour des données de profondeur denses que nous tirons de HOG.

2.3.1 Histogramme des gradients orientés (HOG)

L'histogramme des gradients orientés (HOG) introduit par Dalal et Triggs [44] est actuellement l'une des méthodes les plus performantes et les plus utilisées pour la détection visuelle des personnes. Le procédé considère une fenêtre de détection de taille fixe, subdivisée en une forte densité de réseau uniforme de cellules. Pour chaque cellule, les orientations de gradient sur les pixels sont calculées et recueillies dans un histogramme 1D. L'intuition est que l'aspect local et la forme peuvent être caractérisés par une distribution de gradients locaux sans la connaissance précise de leur position dans la cellule. Les groupes de cellules adjacentes, appelées blocs, sont utilisés pour normaliser localement le contraste. Le descripteur, construit par la concaténation de tous les histogrammes de blocs est alors pris pour entraîner un SVM linéaire. Pour la détection de personnes, la fenêtre de détection est glissée sur l'image à plusieurs échelles. Pour chaque position et échelle, les descripteurs HOG sont calculés et classés avec le SVM entraîné.

Le but des histogrammes des gradients orientés (HOG) [44] est de représenter l'apparence et la forme d'un objet dans une image grâce à la manière dont est réparti l'intensité du gradient ou la direction des contours. Ceci est effectué en divisant l'image en cellules et en calculant pour chaque cellule un histogramme des directions du gradient pour les pixels appartenant à cette cellule. La concaténation de ces histogrammes forme le descripteur.

Normalisation Gamma/couleur : L'extraction du descripteur a été évaluée avec plusieurs représentations couleur des pixels comme le niveau de gris, la couleur RGB et LAB, avec, en option, une normalisation Gamma. Cette normalisation n'a qu'une faible incidence sur les performances et n'est donc pas obligatoire. Les espaces de couleur RGB et LAB donnent des résultats comparables, alors que le niveau de gris réduit les performances. L'utilisation de l'information couleur est donc recommandée dans le cas où celle-ci est disponible.

Calcul des gradients : Les performances du détecteur sont sensibles à la manière dont ont été calculés les gradients, mais la technique la plus simple semble être la meilleure. Dalal et Triggs [44] ont testé le prétraitement de l'image avec un lissage gaussien, puis calculé les gradients en appliquant des filtres dérivatifs différents à différents σ. Les meilleurs résultats sont obtenus en utilisant le filtré dérivatif centré $[-\mathbf{1} \quad \mathbf{0} \quad \mathbf{1}]$, avec σ = 0 (aucun lissage). Pour les images couleurs, les gradients sont calculés séparément sur chaque canal couleur, le gradient ayant la norme la plus grande est gardé.

Calcul des histogrammes : L'image est découpée en plusieurs cellules de petite taille et pour chaque cellule un histogramme est calculé. Chaque pixel d'une cellule vote pour une orientation entre 0 ˚et 180 ˚dans le cas non signé, ou entre 0 ˚et 360 ˚ dans le cas signé. L'étape suivante est la normalisation des descripteurs, afin d'éviter les disparités dues aux variations d'illumination, ainsi que l'introduction de redondance dans le descripteur. Pour cela, les cellules sont regroupées par bloc (concaténation des histogrammes des cellules d'un bloc), le vecteur de valeur du bloc est ensuite normalisé. Les blocs se recouvrent, donc une même cellule peut participer plusieurs fois au descripteur final. Les normalisations possibles du descripteur final sont données dans le tableau 2.1 où $\boldsymbol{v}$ représente l'histogramme d'un bloc.

Tableau 2.1 Les normalisations possibles

L2-Norme	$v \rightarrow \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}}$
L1-Norme	$v \rightarrow \frac{v}{\|v\|_1 + \epsilon}$
L1-Racine	$v \rightarrow \sqrt{\frac{v}{\|v\|_1 + \epsilon}}$

La quatrième norme, L2-hys calcule d'abord v par la L2-norme. L'étape suivante consiste à limiter les valeurs maximales de v à 0.2 et à re-normaliser. Hormis la L1-norme qui n'offre pas de bons résultats, les autres normes L2-norme, L2-hys et L1-racine présentent de bonnes performances et similaires.

Classification: La dernière étape est la mise au point du classifieur supervisé en utilisant les différents descripteurs HOG. On peut faire appel aux différents classifieurs existants. Comme nous l'avons mentionnée plus haut, dans notre application, un SVM à noyau linéaire est utilisé.

2.3.2 Histogrammes des profondeurs orientées

L'Histogramme des Profondeurs Orientées (HOD, Histogram of Oriented Depths) a été introduit pour détecter des personnes dans le cas d'images de profondeur. Cet histogramme HOD suit la même procédure que l'histogramme HOG. Il considère une subdivision d'une fenêtre fixe en cellules, calcule des descripteurs pour chaque cellule et recueille les gradients de profondeurs orientées en histogrammes 1-D. Quatre cellules forment également un bloc afin de recueillir et de normaliser les histogrammes en L2-hys [44], unité de longueur pour atteindre un niveau élevé de robustesse par rapport au bruit. L'intuition est que les changements de profondeur locaux peuvent caractériser la forme 3-D et l'apparence. Les caractéristiques de HOD obtenues sont utilisées pour l'apprentissage d'un classifieur SVM linéaire.

Prétraitement de l'image de profondeur : Tel que présenté précédemment, l'image de profondeur brute est constituée de valeurs qui codent très inégalement la véritable profondeur métrique. Pour les objets éloignés, une différence d'une valeur de profondeur peut correspondre à un saut de plusieurs centimètres. Ceci est d'une importance particulière pour les HOG/HOD car il est connu que les blocs de silhouette qui se trouvent sur les contours d'objets reçoivent des poids très élevés. Plus précisément, il s'agit des blocs qui correspondent aux dimensions de l'hyperplan SVM avec des poids positifs les plus élevés. Par conséquent, nous prétraitons l'image de profondeur en utilisant l'équation 2.1 pour améliorer la séparation plan-fond. Pour la stabilité numérique dans le calcul du gradient, les valeurs de profondeur métrique résultant de M/D_{max} avec $M=100$ correspondant à un gain constant et la distance maximale $D_{max}=20\ m$. Ce prétraitement améliore les contrastes dans les images d'intensité.

Recherche dans l'espace échelle : De nombreuses méthodes de détection visuelle, comme le HOG, utilisent la recherche espace-échelle pour trouver des objets dans une image. Dans le cas du HOD, nous pouvons utiliser les informations de profondeur pour guider ce processus. Le résultat ne sera pas seulement plus efficace, mais constitue une recherche plus précise. Pour améliorer cette recherche, il est nécessaire de créer une technique rapide pour distinguer les échelles compatibles à chaque position dans l'image de profondeur. Dans un premier temps, la hauteur moyenne de l'homme $\boldsymbol{H_m}$ est calculée à partir de l'ensemble de données d'entraînement, dans lequel la position au sol et la hauteur de chaque échantillon sont annotées avec précision. Cette information est ensuite utilisée pour calculer une régression échelle de profondeur comme le montre la figure 2.3 qui suit la relation suivante :

$$S = \frac{F_y.H_m}{d} \cdot \frac{1}{H_w} \qquad (2.2)$$

où F_y est la longueur focale verticale de la caméra IR, $H_m = 1.74\,m$ est la hauteur moyenne mesurée d'une personne et H_w est la hauteur en mètres de la fenêtre de détection à l'échelle 1.

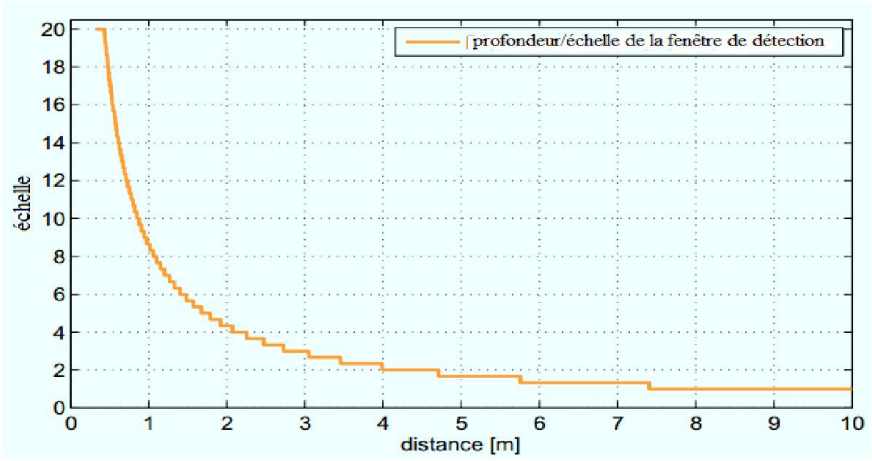

Fig.2.3 Courbe de régression quantifiée qui relie la profondeur à l'échelle de la fenêtre de détection. La courbe est saturée à l'échelle maximum de 20, pour éviter de très grandes fenêtres de détection.

Notons que le terme de gauche de l'équation 2.2 représente la projection d'un demi-plan de hauteur H_m, perpendiculaire à la caméra et situé à une distance d. Pour limiter l'utilisation de mémoire, la valeur de s est quantifiée tout les $1/3$. Nous calculons l'échelle s pour chaque pixel de l'image de profondeur pour générer une carte à échelle d'où nous tirons une liste S de toutes les échelles utilisées. La liste ne contient que les échelles qui sont compatibles avec la présence de personnes dans l'image. Cette méthode évite la prise en compte de plusieurs échelles. Compte tenu de la liste des échelles S qui est calculée une fois pour chaque image, nous pouvons commencer la recherche multi-échelle informée. Les fenêtres dont la profondeur correspond à S sont transmises au SVM. La manière naïve pour résoudre ce problème est de sélectionner une échelle s dans la liste S et de tester si les valeurs de profondeur sous la fenêtre sont compatibles avec s à chaque position espace-échelle. Ce qui implique de faire scanner toute la zone sous la fenêtre de recherche dans toutes les positions et tester si au moins une valeur de profondeur est compatible avec s, une procédure qui est coûteuse en temps de calcul en particulier à grande échelle. En utilisant les images intégrales [14], on propose une solution beaucoup plus rapide capable de tester l'échelle en O(1). Les images Intégrales sont une technique efficace pour calculer la somme des valeurs dans une zone rectangulaire d'une grille. La

valeur de chaque point d'image est la somme de toutes les valeurs ci-dessus et à la gauche du point. La construction de l'image intégrale est elle-même d'une procédure O(N), où N dépend de la taille de l'image d'origine. Le principal avantage d'images intégrales est le calcul d'une zone intégrale avec seulement 4 soustractions. Ici, nous étendons le concept en tenseurs intégrales, images intégrales multicouches avec autant de couches que les échelles de S soumis à la quantification de l'équation 2.2. Chaque couche dans le tenseur intégrale est une image binaire dont les pixels non blancs correspondent à l'échelle de la couche. Ceci permet de tester de manière très efficace si une fenêtre de recherche donnée contient au moins un pixel d'une échelle particulière. La construction du tenseur intégral doit être faite qu'une seule fois par image. Dans la phase de détection, une échelle *s* de *S* est sélectionnée. Ensuite, pour chaque position de la fenêtre de recherche, le test est pris comme une zone intégrale sur la fenêtre de recherche dans la couche du tenseur intégral qui correspond à *s*. Si le résultat est supérieur à zéro, il existe au moins un pixel de profondeur compatible sous la fenêtre et un descripteur HOD est calculé. Sinon, la fenêtre de détection n'est pas considérée et le processus se poursuit. L'algorithme permettant le calcul du HOD donné à la figure 2.4.

Les étapes prises pour avoir le descripteur HOD sont :

- Calcul de gradients horizontaux et verticaux.
- Calcul de l'orientation et l'amplitude du gradient.
- Pour chaque fenêtre de taille 64x128,
  - Diviser l'image en des blocs de 16x16 avec 50% de chevauchement.
  - En total 7x15=105 blocs.
- Chaque bloc doit avoir 2x2 cellules avec une taille de 8x8 pixels pour chacune.
- Quantifier l'orientation du gradient en 9 zones
  - Le vote est l'amplitude du gradient.
  - Interpoler les votes d'une façon bilinéaire entre les centres de zones voisins
- Le vote peut être aussi pondéré avec une gaussienne pour avoir moins d'influence des pixels qui se trouvent sur les bords du bloc.
- Concaténer les histogrammes (descripteur de dimension 105x4x9 = 3780)

Fig. 2.4 Algorithme de calcul du descripteur HOD

2.3.3 Détails de l'implémentation

Calcul du gradient

Les masques utilisés dans les directions x et y sont : $(1 \quad 0 \quad -1)$ et $\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$

Amplitude $A = \sqrt{A_x^2 + A_y^2}$

Orientation $\theta = arctg(\frac{A_y}{A_x})$

Blocs et cellules

-Blocs de 16x16 avec 50% de chevauchement. Au total, on a 7x15=105 blocs.

-Chaque bloc doit avoir 2x2 cellules avec une taille de 8x8 pixels pour chacune. (Fig.2.5).

L'interpolation

-Chaque bloc doit avoir 2x2 cellules avec une taille de 8x8 pixels pour chacune.

-Quantifier l'orientation du gradient en 9 zones (0°-180°)

-Le vote est fait par l'amplitude du gradient.

-Interpoler les votes d'une façon bilinéaire entre les centres de zones voisines.

Comme exemple, prenons θ=85°. La distance aux centres des zones voisines 70° et 90° sont 15° et 5° respectivement. Les rapports seront $\frac{5}{20} = \frac{1}{4}$ *et* $\frac{15}{20} = \frac{3}{4}$ respectivement.

Le vote peut être aussi pondéré avec une gaussienne pour avoir moins d'influence des pixels qui se trouvent sur les bords du bloc.

Les programmes sont écrits en C++ en utilisant la bibliothèque de traitement d'images OpenCV [47]. L'obtention de la carte de profondeur par l'outil logiciel OpenNI [48].

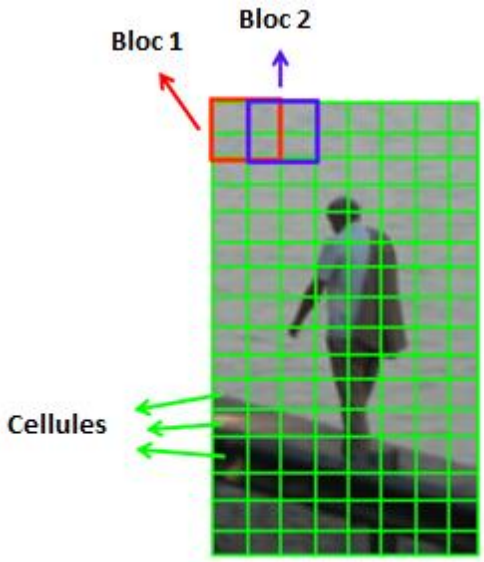

Fig. 2.5 Représentation des blocs et des cellules

Vecteur de caractéristiques final

On concatène tous les histogrammes et le résultat est une matrice 1D de longueur 3780 (Fig.2.6).

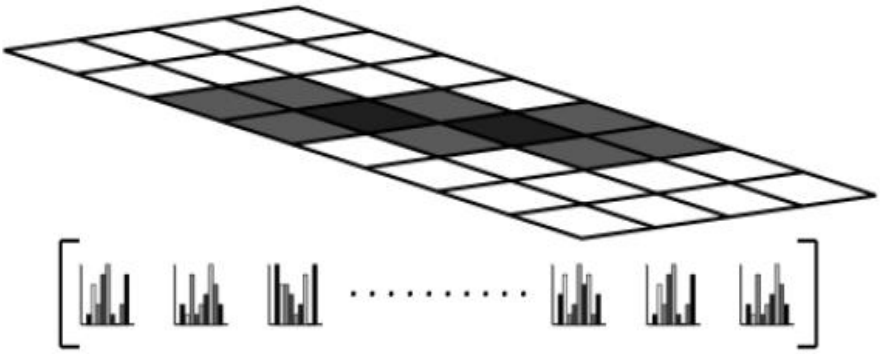

Fig. 2.6 Concaténation de tous les histogrammes, un vecteur de longueur 3780

Apprentissage par SVM et utilisation de l'image intégrale

Après avoir calculé le vecteur de caractéristiques, on passe à l'apprentissage par SVM. Pour ce faire, prenons des images de profondeurs et choisissons des fenêtres là où il y a des personnes comme des exemples positifs. Pour les exemples négatifs, nous prenons des fenêtres qui ne contiennent pas de personnes. La base de données utilisée est celle de Spinello [49]. Nous associons aux exemples positifs, la valeur +1 et aux exemples négatifs, la valeur -1.

Notre base de données pour l'entraînement du SVM est constituée de 1030 exemples positifs et 5000 exemples négatifs qui sont choisis aléatoirement.
L'image intégrale permet de tester de manière très efficace si une fenêtre de recherche donnée contient au moins un pixel d'une échelle particulière. La construction du tenseur intégral doit être faite qu'une seule fois par image.
Dans la phase de détection, une échelle *s* de *S* est sélectionnée. Ensuite, pour chaque position de la fenêtre de recherche, le test est pris comme une zone intégrale sur la fenêtre de recherche dans la couche du tenseur intégral qui correspond à s. Si le résultat est supérieur à zéro, il existe au moins un pixel de profondeur compatible sous la fenêtre et un descripteur HOD est calculé. Sinon, la fenêtre de détection n'est pas considérée et le processus se poursuit.

2.3.4 Résultats et discussion

Procédure expérimentale

Pour faire l'apprentissage, nous avons utilisé la base d'image de Spinello disponible sur la page web des auteurs [49]. La base d'images a été prise dans un hall d'un grand restaurant universitaire à l'heure du déjeuner. Un ensemble d'images supplémentaire a été recueilli dans un autre bâtiment de l'université utilisé uniquement pour générer des échantillons de fond. La base d'images a été annotée manuellement pour inclure la zone de délimitation en espace image en profondeur 2D et l'état de visibilité des personnes (entièrement visible / partiellement occlus). Un total de 1 648 cas de personnes en 1088 trames a été étiqueté.
Comme mesures d'évaluation, nous déterminons les taux de précision et de rappel. Les détections sont comptées comme des vrais positifs, si la zone de délimitation se chevauche avec une personne étiquetée manuellement par plus de 60% pour tenir compte des inexactitudes métriques dans l'annotation et la détection. Nous ne comptons pas de vrais positifs ou de faux positifs en cas une détection correspond à une annotation d'une personne partiellement occlus. La base d'apprentissage est composée de 1030 échantillons de données de profondeur de personnes qui sont

également en miroir sur l'axe horizontal et 5000 échantillons négatifs qui ont été choisis au hasard parmi l'ensemble des données de fond.

$$Précision = \frac{T_p}{T_p + F_p} \quad (2.3)$$

$$Rappel = \frac{T_p}{T_p + F_n} \quad (2.4)$$

Où:

T_p : Le nombre des vrais positifs

F_p : Le nombre des faux positifs

F_n : Le nombre des faux négatifs

Performance

Nous comparons notre détecteur HOD avec d'autres techniques basées sur la profondeur et des techniques visuelles qui utilisent les images RGB.

Étant donné l'importance de la quantification de la profondeur pour les données Kinect, nous évaluons deux variantes de HOD: HOD11 qui considère la gamme complète (11 bits) de l'information de profondeur disponible à partir du capteur et HOD8 qui utilise une gamme réduite (8 bits).

Les résultats du Tableau 2.2 montrent clairement que le HOD11 surpasse HOD8 sur toute la plage de précision/rappel: 3 bits supplémentaires pour coder la profondeur aide à lever l'ambiguïté des gens de fond. Cela est également vrai pour toutes les opérations de prétraitement sur les données de profondeur.

Une question fondamentale dans le contexte des données RGB-D est la contribution de l'information de profondeur par rapport aux techniques de détection purement visuelles. Pour évaluer cette question, nous considérons les performances du détecteur HOG visuel qui détecte les personnes dans les images RGB. Comme on peut le voir sur le Tableau 2.2, le HOG sous-performe par rapport à HOD11, avec un taux de précision de 73% pour HOG et 85% pour HOD11. La principale raison de ces résultats modestes est liée à des problèmes d'éclairage. L'environnement de l'ensemble de données n'est pas éclairé d'une façon optimale. Les régions d'arrière-plan, avec de la lumière du soleil directe, résultent à des zones d'image saturées et à un

mauvais contraste. Les résultats démontrent la nécessité de systèmes de détection de personnes qui travaillent dans des gammes de conditions qui sont plus larges que ceux des approches de détection purement visuelles et motivent l'utilisation d'informations de profondeur pour cette tâche.

Les performances de calcul du détecteur HOD sont également évaluées. Nous comparons le nombre d'échelles que le HOD traite par image à l'aide de la recherche multi-échelle informée par rapport à la méthode HOD non informée (notée HOD-). HOD- utilise une recherche pyramidale avec un incrément d'échelle de 5% quel que soit le contenu de l'image. Ce qui est différent au HOD où l'échelle est une fonction de la profondeur et change pour chaque nouvelle image de profondeur. Nous affirmons une diminution de près de trois fois le nombre d'échelles qui sont recherchées sur toutes les images testées. Cela conduit à une accélération approximative de 3 fois du temps de traitement par image entre HOD et HOD-. L'algorithme est en mesure de traiter le flux de données de profondeur de la Kinect (640 × 480 pixels à 10 images par seconde) en temps réel sur un ordinateur portable Core I5 à 2,53 GHz.

Les résultats qualitatifs du détecteur HOD sont présentés dans la figure 2.7. La figure illustre plusieurs personnes détectées à des distances différentes avec différents occlusions partielles et dans différentes conditions. Ces tests ont été faits dans un milieu fermé. Quand une personne est détectée, elle est encadrée en temps réel.

Tableau 2.2 Taux de précision et de rappel

Approche	**Précision (%)**	**Rappel (%)**
HOD8	73	70
HOD11	85	80
HOG	80	83

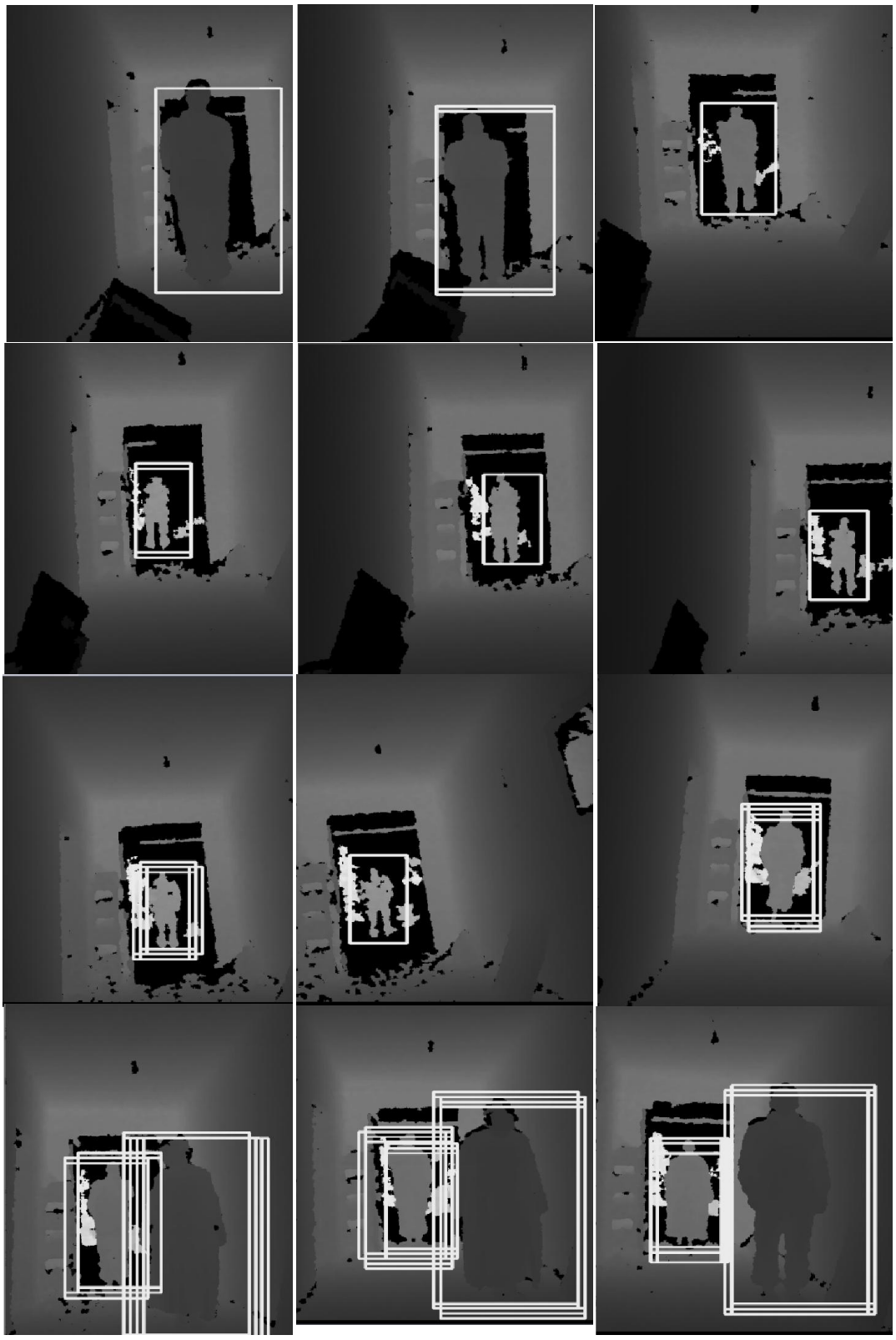

Fig. 2.7 Résultats qualitatifs de la détection de personnes avec le HOD

2.4 Conclusion

Dans ce chapitre, nous avons introduit le HOD, une approche pour le problème de la détection de personnes dans les images de profondeur. Nous avons décrit des informations clés sur les caractéristiques des données Kinect, le capteur utilisé dans les expériences, ce qui nous a guidés dans le développement des méthodes proposées. Le HOD, qui représente l'histogramme des profondeurs orientées, code localement la direction des changements de profondeur et repose sur une recherche multi-échelle de profondeur informée qui conduit à une accélération de 3 fois le processus de détection. Le résultat est un détecteur de personnes qui réalise un taux d'égale erreur de 85% dans une plage de presque deux fois plus grande que celle spécifiée par le constructeur du capteur. Nous avons en outre mené des expériences comparatives d'analyser la contribution des données de profondeur par rapport aux méthodes purement visuelles. Le HOD implémenté sur un ordinateur portable Core I5 à 2,53 GHz surpasse le HOG et il est exécuté à 10 tps.

Chapitre 3

Identification de postures et de gestes

3.1 Introduction

Avec l'avènement récent des capteurs de profondeur, les applications de la reconnaissance des gestes en ligne ont été diversifiées et multipliées. Pour ce faire, la robustesse du processus de reconnaissance est nécessaire pour faire face aux données bruitées de ces capteurs et tenir compte de la vitesse d'exécution de la reconnaissance pour le fonctionnement de l'interface communication homme-machine.

Ainsi, dans ce chapitre, nous présentons une méthode de reconnaissance de positions et de gestes d'un humain en temps réel à partir de squelettes bruités extraits du capteur de profondeur Kinect. Chaque pose est décrite en utilisant une représentation angulaire des articulations du squelette. Ces descripteurs servent à identifier des poses clés à travers un classifieur SVM multi-classes. Ces poses clés sont utilisées pour reconnaitre les gestes à l'aide d'une méthode de décision de type forêt.

3.2 Principe de la méthode utilisée

La reconnaissance de positions et de gestes d'un humain en temps réel que nous avons développée est inspirée du travail de Miranda et al. [42]. La méthode suit les trois phases classiques à savoir le choix des descripteurs, la méthode d'apprentissage et la décision (Fig.3.1). En effet, les descripteurs de positions doivent représenter de façon concise une pose du corps humain. La méthode d'apprentissage multi-classes doit présenter les poses clés de façon robuste. La décision doit permettre la reconnaissance des gestes en tenant compte des contraintes de temps et de vitesse.

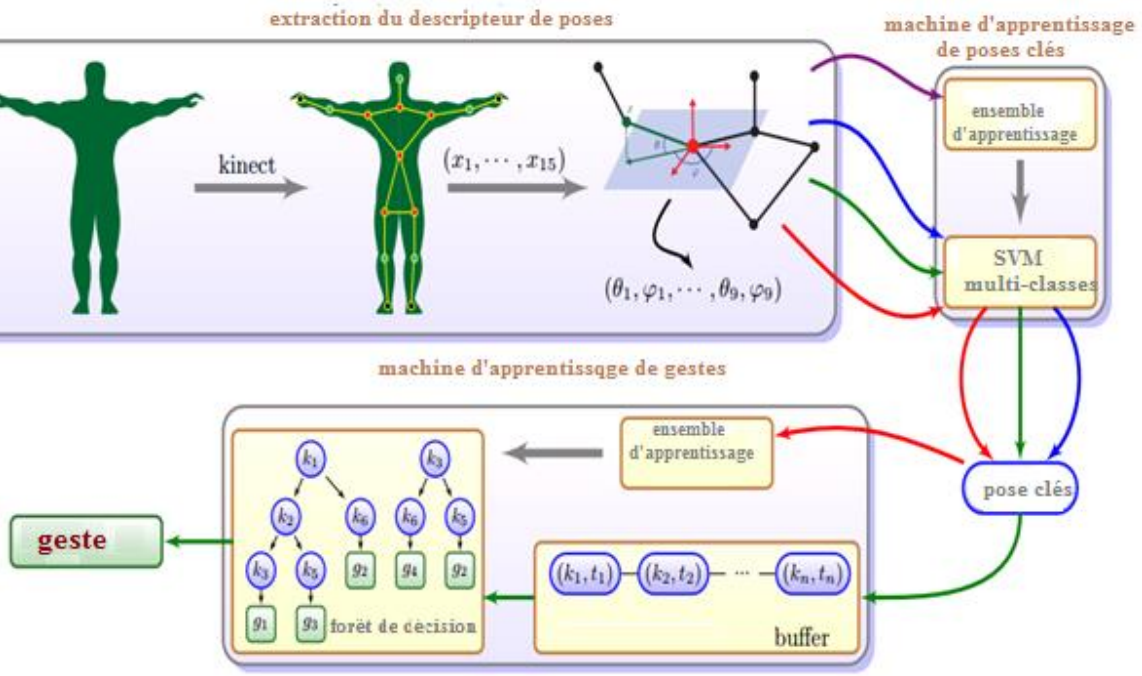

Fig.3.1 Schéma résumant la reconnaissance de poses et de gestes [42]

Comme tout cela sera détaillé plus bas, nous procédons d'abord à l'extraction du squelette de l'humain. Ensuite, nous convertissons l'ensemble des points représentant les nœuds du squelette en une représentation angulaire pure réduite, inspirée de la méthode de Raptis et al. [43]. Elle est invariante à l'orientation du capteur et réduit la redondance et la dimensionnalité, tout en préservant les informations pertinentes pour la classification des poses clés.

En ce qui concerne l'apprentissage des poses clés, nous utilisons une approche SVM multi-classes [50]. Lors de cette phase d'apprentissage, plusieurs exemples de poses clés ont été définies pour en constituer une base d'apprentissage. Cette base d'apprentissage est utilisée pour construire plusieurs classifieurs SVM binaires qui reconnaissent de façon robuste les poses clés, dans une approche un contre-tous.

Après l'apprentissage, nous passons à la phase de reconnaissance en nous basant sur les poses clés et en utilisant une forêt de décision optimisée pour permettre la recherche efficace pour n' importe quelle séquence de poses clés qui composent un geste. Pour chaque performance d'un geste, les poses clés sont accumulées dans une mémoire tampon circulaire en vérifiant, dans la forêt de décision, si la séquence complète un geste connu. Cela évite la nécessité d'une pose initiale-neutre. Les nœuds de la forêt peuvent éventuellement considérer les contraintes temps-vitesse.

Même lorsque différents utilisateurs effectuent le même geste avec une durée différente entre les poses clés, la forêt de décision apporte une solution efficace et robuste au problème d'alignement temporel.

3.3 Apprentissage des poses clés

Les méthodes de la reconnaissance des gestes à travers les poses clés sont fortement dépendantes de la robustesse de la classification des poses, exigeant une efficacité afin de fournir une performance en temps réel. Pour résoudre ce problème de classification, nous proposons une approche d'apprentissage supervisé, où les poses clés d'apprentissage sont obtenues à partir de l'utilisateur. L'utilisateur peut, à tout moment, rajouter des données d'apprentissage supplémentaires pour corriger et améliorer la robustesse du classifieur, tout en gardant son efficacité.

Nous construisons un tel classifieur en utilisant une composition multi-classe des Machines de Vecteur de Support (SVM) binaires dont la formulation est bien adaptée pour répondre les exigences attendues.

Dans ce qui suit, nous décrivons quelques aspects fondamentaux de l'approche SVM multi-classes que nous avons adoptée, ainsi que la représentation squelette angle d'articulation que nous avons développée.

3.3.1 Représentation angle d'articulation

La représentation de squelette doit être invariante à l'orientation du capteur et la translation globale du corps. Elle doit également être en mesure de minimiser les problèmes avec les variations de la taille du squelette de différents individus tout en capturant toutes les informations pertinentes de la position du corps.

Les données squelette brutes de la Kinect sont, pour chaque trame, une séquence graphique de 20 nœuds [6], où chaque nœud a sa position géométrique représentée comme un point 3D dans un système de coordonnées cartésiennes (Fig.3.2).

Les articulations adjacentes au torse sont appelées articulations de premier ordre alors que les articulations adjacentes aux articulations du premier ordre sont classées comme articulations de second ordre. Les articulations de premier ordre comprennent les coudes, les genoux et la tête tandis que les extrémités, les mains et les pieds, sont considérées comme articulations de deuxième ordre.

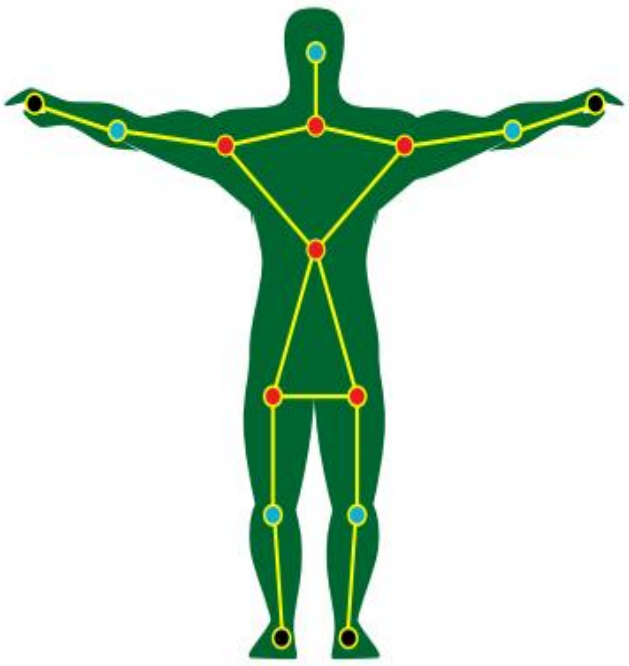

Fig.3.2 Squelette Kinect, en rouge les articulation du torse, en vert les articulations de 1er ordre, en noir les articulations de 2ème ordre

Des poses différentes du corps sont essentiellement obtenues par rotation des articulations de premier et de deuxième ordre. Notons que chaque mouvement d'une articulation a deux degrés de liberté, un angle zénith θ et un angle d'azimut φ (Fig.3.3).

La distance entre les articulations adjacentes est toujours constante pour un individu donné.

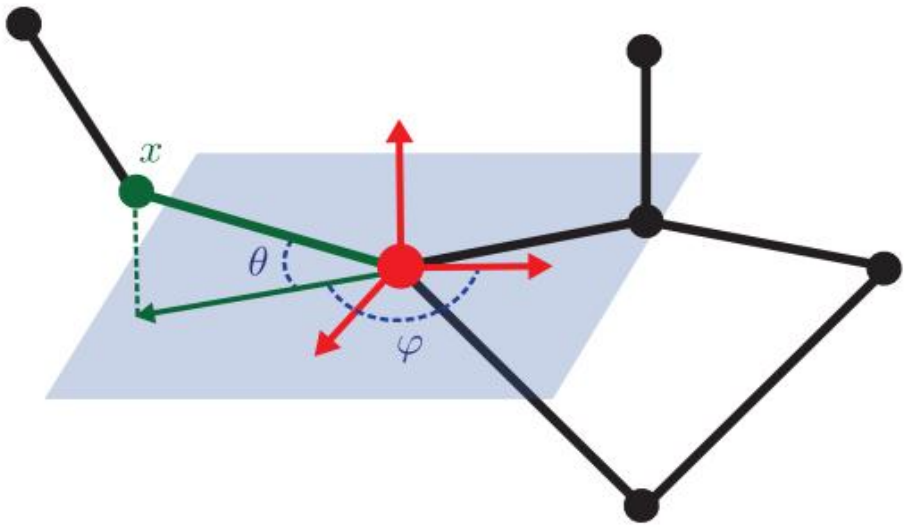

Fig.3.3 Représentation par angle d'articulation

Dans le travail de Raptis et al. [43], une représentation angle d'articulations, simple est proposée en écrivant la position $X_l \in R^3$ d'une articulation l en coordonnées sphériques locales, en omettant la distance radiale. Pour cela, une base de torse est estimée par l'application d'une Analyse en Composantes Principales (ACP) à une matrice de torse 7×3 remplie avec les positions des nœuds de torse. Ensuite, les coordonnées sphériques de chaque articulation du premier ordre sont calculées comme une translation de cette base de torse à l'articulation. Cependant, la même base de torse est utilisée comme référence pour convertir les articulations du deuxième ordre, ce qui conduit à une description non locale des angles. Nous utilisons la même base de torses pour les articulations du premier ordre, mais nous améliorons la représentation des articulations de deuxième ordre en considérant les rotations de la base orthonormée du torse $\{u, r, t\}$. [42]

Repère lié au torse par application de l'ACP

Nous observons que les points du torse humain défini par 7 nœuds squelettiques comme illustré sur la figure 3.2 ne présentent que rarement un mouvement indépendant. Ainsi, le torse peut être traité comme un corps rigide allongé verticalement. Pourtant, en raison des bruits dans le système de détection de profondeur, nous observons que les points de torse individuels, en particulier les épaules et les hanches, peuvent présenter des mouvements non réalistes que nous préférons limiter plutôt que de les propager par une représentation relative. Par conséquent, nous visons à traiter le torse comme un corps rigide avec tous ses points contribuant à l'estimation de sa position. Ensuite, nous utilisons cette estimation pour représenter le reste du squelette humain de manière relative.

Nous calculons les composantes principales pour les points de torse, c'est à dire, une base 3D orthonormée à la suite de l'application de l'ACP à la matrice de torse de dimension 7 par 3. La première composante principale u est toujours alignée avec la plus grande dimension du torse et nous pouvons canoniquement l'orienter vers le haut ou vers le bas car, dans la plupart des cas, il n'est pas prévu que le torse de la personne se tienne à l'envers par rapport au capteur. En revanche, pour la deuxième composante principale r, alignée avec la ligne qui relie les épaules, l'orientation n'est pas si facilement déduite et, ici, nous devons compter sur l'orientation du squelette "gauche-droite" inférées par la l'algorithme de poursuite du squelette. Enfin, le dernier axe de la base orthonormée est calculé comme un produit vectoriel des deux premières composantes principales, à savoir, $t = u \wedge r$. Nous appelons la base résultante $\{u, r, t\}$ le repère du torse.

Le repère du torse est bien aligné avec nos objectifs précédemment énoncés. C'est une base exceptionnellement robuste et fiable pour un système de coordonnées sur la base de l'orientation du corps humain. Bien qu'il soit dépendant de la position de la caméra, des points représentés dans un système de coordonnées qui est dérivé de la base de torse peuvent être totalement invariants au capteur. Il réduit les 7 trajectoires 3D de la spécification originale du problème à un nouvel ensemble de signaux dont le but est de décrire seulement l'orientation 3D de la base orthonormée

résultante. Nous détaillons dans ce qui suit un ensemble de caractéristiques simples qui décrivent de manière intuitive et robuste la pose du corps.

Joints du premier degré

Nous notons tous les joints adjacents au torse comme joints de premier degré. Ceux-ci incluent les coudes, les genoux et la tête. Nous représentons ces points par rapport à l'articulation adjacente au niveau du torse dans un système de coordonnées provenant du repère de torse. Considérons la partie centre de la figure 3.4 où *LE*, le coude gauche, est représenté par rapport à *LS*, l'épaule gauche. Tout d'abord, nous translatons le repère du torse, (u, r, t) au point *LS* et nous construisons un système de cordonnées sphériques de telle sorte que l'origine est centrée en *LS*, l'axe du zénith étant u et l'axe d'azimut r.

Ainsi, la position de *LE* est décrite par:

R : la distance de *LE* par rapport à l'origine du repère,

θ : l'angle entre u et $\overrightarrow{(LS, LE)}$, et

φ : l'angle entre r et $\overrightarrow{(LS, LE_p)}$ ou LE_p est la projection de *LE* sur le plan défini par son normal u.

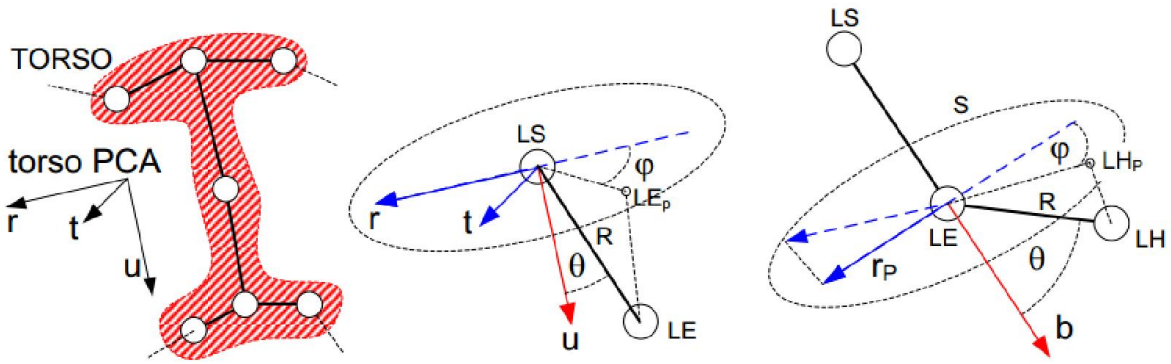

Fig. 3.4 Illustration de la base du torse obtenue par l'ACP, et son utilisation dans la définition des systèmes de coordonnées sphériques des articulations de premier et de deuxième degré [43]

Puisque la distance entre les articulations est normalisée et constante, nous ignorons *R*. Ainsi, en utilisant ce modèle de représentation, chaque joint du premier degré est représenté avec deux angles $\{\theta, \varphi\}$.

Joints de deuxième degré (par Raptis)

Notons, comme joints du second degré, les extrémités du corps. Ceux-ci comprennent les mains et les pieds. Le vecteur le plus descriptif associé à un joint de deuxième degré est l'os qui relie le joint du premier degré adjacent à son joint adjacent appartenant au torse. Par exemple, comme le montre la partie droite de la figure 3.4, un vecteur b est un excellent candidat pour la direction du zénith d'un système de coordonnées sphériques avec une origine, le coude gauche *LE*. Désignons par *LH*, l'articulation de la main gauche. La position de *L*H est décrite par:

R : la distance de LE par rapport à l'origine du repère,

θ : l'angle entre b et $\overrightarrow{(LE, LH)}$, et

φ : l'angle entre r_p , la projection de r sur le plan S ayant comme normal b, et $\overrightarrow{(LE, LH_p)}$ ou LH_p est la projection de LH sur le plan S.

Comme précédemment ; puisque la longueur de l'os de l'avant-bras est normalisée et constante, nous ignorons *R*. Ainsi, notre modèle représente chaque joint de deuxième degré en utilisant deux angles $\{\theta, \varphi\}$. Les conséquences sont équivalentes à celles des articulations du premier degré à une différence notable. Bien que l'inclinaison θ pour les joints de deuxième degré soit un descripteur extrêmement robuste, leur azimut ne l'est pas. Etant donné que l'origine du système de coordonnées sphériques ne fait pas partie du corps rigide qui définit la base du torse, l'orientation de r dépend de l'orientation du torse qui fait introduire des bruits sur φ. Dans [43], l'auteur dans ses expériences a confirmé que cet effet est plutôt doux et ne pose pas un défi pour les étapes restantes du classifieur. Par la suite, dans notre cas, ce problème est résolu par l'application de représentation de Miranda [42].

Notons que les vecteurs b et r peuvent être orientés de telle façon que $b.r = 1$, ce qui fait la projection r_p est un point. Bien que ce soit peu probable, n'importe quel petit angle entre les vecteurs b et r est susceptible d'introduire des niveaux de bruit en raison de l'instabilité de r_p. Bien que ce problème peut être résolu de plusieurs manières, nous avons observé dans notre application que le cas $b.r \approx 1$ se produit rarement lorsque r est choisi

comme une référence d'azimut. On notera qu'au lieu de r nous aurions pu utiliser u ou t ou toute combinaison linéaire de ces vecteurs avec un large impact sur la performance finale. En ce qui nous concerne, nous avons choisi r parce que sa sélection atténue suffisamment cet effet.

Joints de deuxième degré (par Miranda)

Nous utilisons la même base de torses pour les articulations du premier ordre, mais nous améliorons la représentation des articulations de deuxième ordre en considérant les rotations de la base orthonormée du torse $\{u, r, t\}$.

Plus précisément, soit v, w les vecteurs supportant les os adjacents à une articulation. Par exemple, v serait le vecteur défini par l'épaule droite et le coude droit et w le vecteur entre le coude droit et le poignet droit. Pour définir une base locale pour la main droite, nous faisons tourner la base du torse $\{u, r, t\}$ par un angle $\beta = v.r$ autour de l'axe $b = v \wedge r$. Cette base est translatée au niveau du coude droit et les coordonnées sphériques de la main droite sont calculées comme suit:

θ : L'angle entre v et w .

φ : L'angle entre le nouveau t et la projection de w sur le plan orthogonal à v .

Si v et w sont colinéaires, nous mettons $\varphi = 0$ puisque l'azimut n'est pas définie. Les articulations du deuxième ordre sont ainsi construites en utilisant des variantes de la base de torse.

Enfin, chaque position d'articulation X_l est représentée en utilisant un pair d'angles sphériques (θ, φ) qui la décrit dans un système de coordonnées sphériques défini localement. Considérons un squelette avec neuf articulations, une représentation d'une pose du corps est le vecteur descripteur :

$$v = (\theta_1, \varphi_1, \dots\dots\dots, \theta_9, \varphi_9) \in R^{18}$$

3.3.2 Formulation SVM multi-classes

Le classifieur cherche des similitudes avec les poses clés de référence dans un ensemble prédéfini $K = \{c^1, c^2, \dots\dots, c^{|K|}\}$. Ces classes de poses clés

seront utilisées pour construire des représentations de gestes. Pendant l'apprentissage des poses clés, nous créons un ensemble d'apprentissages en fournissant plusieurs exemples de chaque pose clé. Dans nos expériences, 40 exemples par pose clé ont été utilisés. La machine d'apprentissage SVM multi-classes a, comme données, l'ensemble d'apprentissage $T = \{(v^1, c^1), (v^2, c^2), \ldots\ldots\}$ où chaque pose clé v est entrainée par l'utilisateur pour une étiquette spécifique c. Plus précisément, chaque vecteur $v^i \in R^{18}$ est le descripteur de la pose clé entrainée, tandis que $c^i \in \{1,2, \ldots, |K|\}$ est un entier identifiant la classe de la pose clé.

3.4 Reconnaissance des gestes par la forêt de décision

Un geste peut être représenté par une courbe continue dans le temps et dans l'espace de toutes les poses réalisables du corps. Cependant, la plupart des gestes peuvent être identifiés par une séquence de quelques poses clés, qui peut être considérée comme un échantillonnage de la courbe du mouvement continu.

La reconnaissance gestuelle utilise également une approche d'apprentissage supervisé. En effet, au lieu des SVM, l'apprentissage des gestes est structuré dans une forêt de décision qui reconnaît efficacement les gestes, analyse la séquence des poses clés faite par l'utilisateur en temps réel.

3.4.1 Définition d'un geste

Nous avons présenté des gestes comme des séquences $g = \{k_1, k_2, \ldots\ldots, k_{n_g}\}$ de poses clés $k_i \in K$. En effet, un geste est généralement identifié par une petite séquence de deux à cinq poses clés. Par exemple, un geste composé d'ouverture des bras et après un claquement des mains peut avoir besoin aussi un peu plus que 4 poses clés comme le montre figure 3.5. Souvent, des exécutions légèrement différentes de la même classe de geste peuvent conduire à des séquences de poses clés différentes.

Une représentation simple d'un geste serait une séquence unique $\{k_1, k_2, \ldots\ldots, k_{n_g}\}$ de poses clés composant le geste g . Bien qu'elle soit efficace, cette approche peut ignorer une autre façon d'effectuer le même geste.

Dans ce travail, nous limitons l'ensemble des gestes de telle sorte que la séquence définissant un geste ne peut être étendue à une séquence plus longue définissant un autre geste, c'est à dire les gestes sont irréductibles. Un geste est alors défini de manière unique par un ensemble de séquences de poses clés qui sont différentes séquences caractérisant le même geste exécuté par différents utilisateurs.

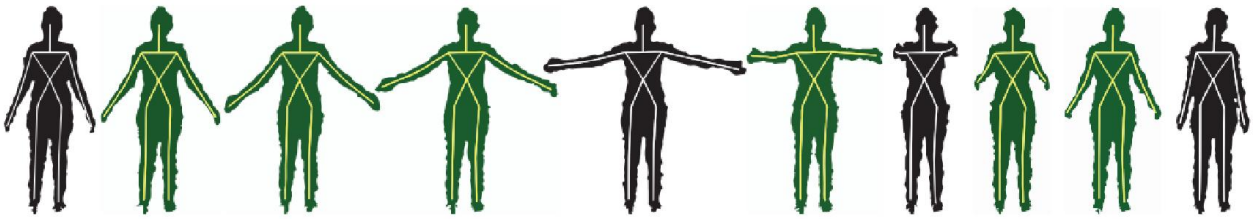

Fig.3.5 Représentation d'un geste par des poses clés (en noir)

Cela diffère légèrement de la courbe d'action [26], en omettant les probabilités de transition entre poses clés mais permettant une représentation par une forêt de décision, ce qui permettra d'éviter la nécessité d'une posture neutre/initial. Nous optimisons le schéma de la forêt de décision pour identifier efficacement les gestes en temps réel.

3.4.2 Reconnaissance de gestes

Étant donné un ensemble d'apprentissage de geste composé de séquences de poses clés, nous construisons une forêt dont les nœuds représentent les poses clés et dont les feuilles sont associées à des gestes. Chaque chemin dans un arbre de la forêt, de la mère d'une feuille g à la racine, représente une séquence possible pour le geste g . Ainsi, chaque arbre enraciné à une pose clé k code tous les gestes dont la pose clé finale est k , tandis que chaque feuille contient un identifiant de gestes. Notons qu'il existe, au plus autant, d'arbres qu'il ya de poses clés.

Pendant la phase de reconnaissance, les classifieurs de poses clés essaient de reconnaître les poses clés effectuées par l'utilisateur. Puisque nous n'exigeons pas une pose neutre/initiale, nous accumulons les poses clés dans une mémoire tampon circulaire B des dernières poses clés identifiées.

Nous évitons de répéter une pose clé en B en vérifiant si la dernière pose clé ajoutée est identique à une pose clé nouvellement détectée.

Chaque fois qu'une pose clé k est reconnue, il est inséré dans B et une nouvelle recherche est lancée à la racine de l'arbre représentant les gestes qui se terminent par k . Nous recherchons dans l'arbre par itération inverse du tampon à partir de k. Si l'élément précédent dans B, c'est à dire, la pose clé détectée précédemment est un enfant du nœud courant, la recherche continue. Sinon, la recherche échoue, car il n'y a pas de séquence de pose clé entrainée qui est un suffixe de la dernière séquence de poses clés effectuée. Si la recherche atteint une feuille, le geste stockée dans cette feuille est reconnu et signalé. En pratique, le tampon circulaire B n'a pas besoin d'être vidé, mais nous exigeons qu'il ait suffisamment d'espace pour contenir la plus grande séquence.

Le choix de stocker les gestes à l'envers dans la forêt de décision simplifie le travail de reconnaissance. De cette façon, la recherche est effectuée dans un seul arbre, évitant les retards de la programmation dynamique. Dans la pratique, les arbres sont relativement minces, c'est-à-dire, les séquences qui ne correspondent pas échouent rapidement. En outre, cela ne nécessite pas de savoir quelle pose clé de B est la première pose clé du prochain geste puisque nous n'utilisons que la dernière pose pour lancer la recherche.

Nous soulignons que deux séquences de poses clés différentes peuvent être marquées comme même geste. La figure 3.6 montre un exemple d'une simple forêt avec six poses clés et cinq gestes reconnaissables. Notons comment deux séquences différentes de poses clés sont assignées au même geste g_2 . Cette fonction est utile dans des applications, par exemple, quand un geste effectué avec la main droite est considéré comme étant le même que celui réalisé avec la main gauche. En outre, il est possible d'effectuer le même geste avec un passage à travers des séquences des poses clés légèrement différentes.

Enfin, notre formulation ne permet pas de sous-gestes de gestes, pour éviter toute ambiguïté. Toutefois, si les sous-gestes doivent être identifiés, on peut facilement adapter notre méthode représentant un sous-geste comme un nœud intérieur de l'arbre.

3.4.3 Contraintes de temps

Jusqu'à présent, la vitesse d'exécution du geste n'est pas prise en considération, bien qu'elle puisse être importance dans de nombreuses applications telles que les mouvements de danse Raptis et al. [43]. Dans notre représentation de gestes, nous pouvons inclure l'intervalle entre les poses clés consécutives en tant que vecteur de temps $[t_1, t_2, \quad , t_{n-1}]$ associé au tampon circulaire *B*.

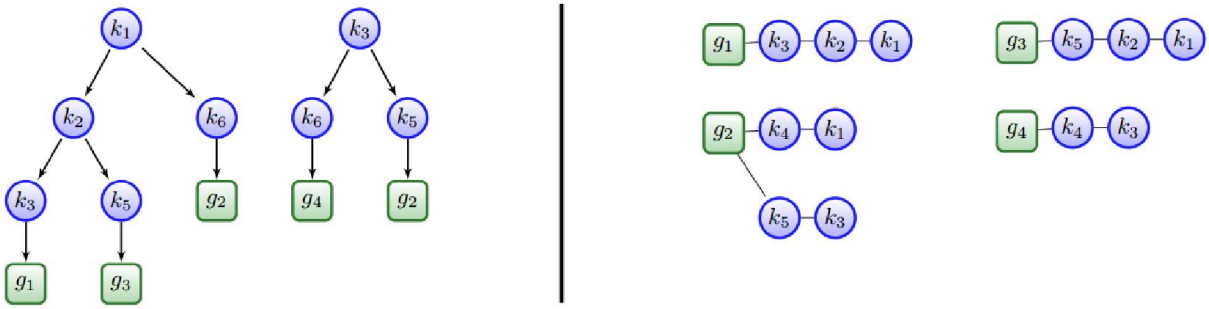

Fig. 3.6 Exemple d'une machine d'apprentissage de gestes : une forêt contenant cinq poses clés (à gauche) ; et les quatre gestes représentés par la forêt (à droite)

Ainsi, la même séquence de poses clés réalisée à des vitesses différentes peut être considérée comme des exécutions de gestes appartenant à des classes différentes ou même pas considérée, du tout, comme un geste.

Nous stockons, dans chaque feuille de la forêt, un ou plusieurs vecteurs de temps des gestes partageant la même séquence de poses clés correspondantes. Deux gestes avec la même séquence de poses clés, mais ayant des vecteurs intervalles de temps différents conduiraient à des feuilles jumelles. Lorsque nous recherchons des gestes dans la forêt de décision, nous choisissons ou rejetons un geste sur la base des vecteurs temps avec deux critères.

Soit t_i un vecteur de temps mémorisé dans une feuille représentant le geste g_i et t le vecteur de temps correspond à la performance de l'utilisateur actuel. Si $\| t_i - t \|_\infty > T$ pour tous les vecteurs de temps mémorisés avec g_i, où T est un petit seuil, alors g_i est rejeté. Parmi tous les gestes non-rejetés, le geste g_i qui minimise $\| t_i - t \|_1$ est choisi comme le geste reconnu.

3.5 Résultats et discussion

Nous présentons dans cette section les expériences que nous avons effectuées pour valider la robustesse et évaluer l'exécution de la méthode proposée.

3.5.1 Procédure expérimentale

Pour évaluer la robustesse de notre machine d'apprentissage de poses clés, nous avons conçu un ensemble de pose clé K composé de 20 poses clés pour être utilisées dans tous les tests. Un exemple de chaque catégorie de ces poses clés est illustré à la figure 3.7. A noter que le squellette est obtenu par l'outil logiciel Kinect SDK [45].

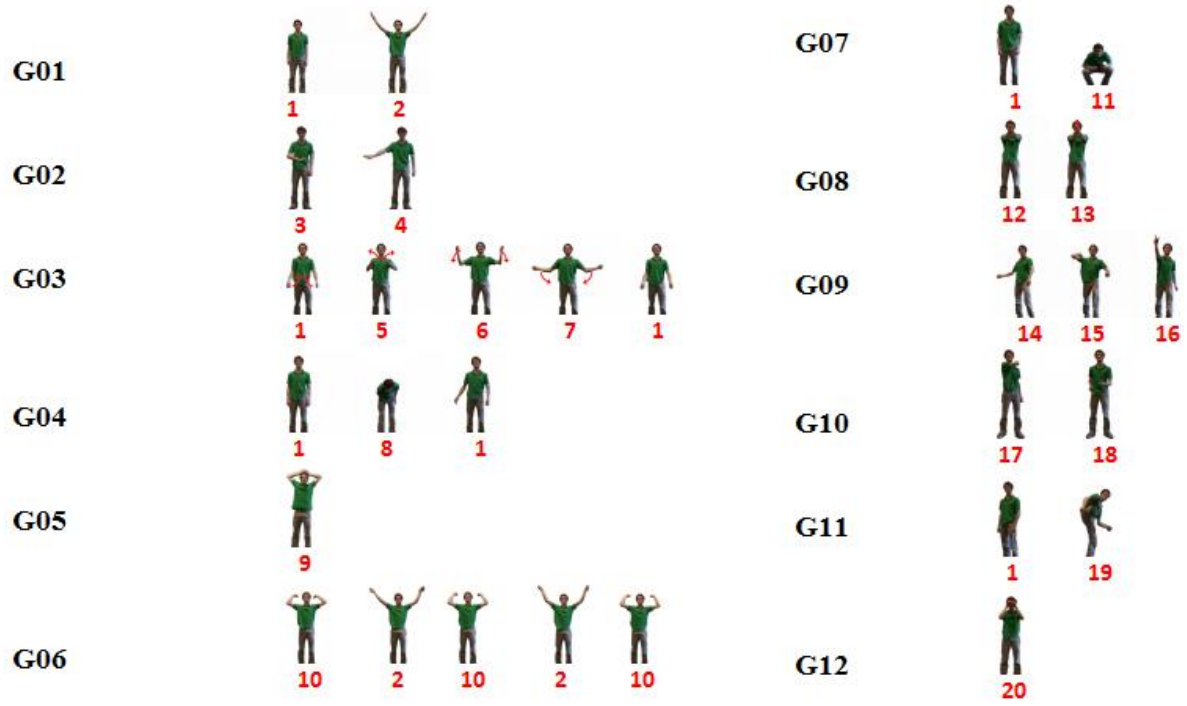

Fig.3.7 Les poses clés de référence pour l'ensemble de l'apprentissage T

Notons que nous nous sommes concentrés principalement sur les poses des membres supérieurs qui sont plus pertinents pour les interfaces utilisateur. Pour créer l'ensemble T des poses clés de d'apprentissage, on a demandé à 10 personnes d'effectuer environ 40 exemples de chaque pose clé, résultant dans environ 8000 exemples de poses clés.

Ensuite, nous avons conçu un ensemble G de 12 gestes, comme indiqué dans la figure 3.7. Nous avons demandé à chaque personne d'effectuer 10 fois chaque geste, et capturé les séquences de poses clés. Nous avons également considéré les contraintes de temps pour les gestes G05 et G12 pour valider notre formulation. Dans les gestes G05 et G12, les poses k09 et k20 doivent être conservée pendant 1 seconde pour caractériser ce geste.
Notons que K restreint l'ensemble des gestes reconnaissables G à toutes les combinaisons finies de poses clés de K. Ainsi, la conception de K doit prendre en compte l'ensemble des gestes reconnaissables souhaités G.

3.5.2 Reconnaissance de poses clés

Nous avons demandé à 10 individus d'effectuer toutes les poses clés pour évaluer le taux de reconnaissance de nos classifieurs. Chaque individu a effectué chaque pose clé 5 fois, les résultats sont présentés dans le Tableau 3.1.

Tableau 3.1 Résultats de la reconnaissance des poses clés

Identifiant de la pose clé	Nombre de poses clés reconnues	Total (%)
1	48	**96**
2	47	**94**
3	46	**92**
4	45	**90**
5	46	**92**
6	47	**94**
7	47	**94**
8	40	**80**
9	48	**96**
10	47	**94**
11	49	**98**
12	48	**96**
13	45	**90**
14	41	**82**
15	44	**88**
16	42	**84**
17	44	**88**
18	45	**90**
19	40	**80**
20	49	**98**

La machine d'apprentissage de poses clés était capable de reconnaître les poses clés des utilisateurs dans la plupart des cas, atteignant un taux de reconnaissance moyenne de 90,8%. Même dans des poses similaires, comme

k6 et k10, la machine a réussi à classer la vraie pose dans la plupart des exemples. Nous avons observé que la plupart des échecs sont dans des poses difficiles pour le suivi du squelette, où l'image de profondeur souffre d'occlusion, comme la pose k8.

Tableau 3.2 Résultats de la reconnaissance des gestes

Geste	Identificateur	Taux de reconnaissance
Augmenter le volume	G1	82%
Accédez au menu suivant	G2	90%
Mouvement circulaire	G3	85%
Mettre fin	G4	70%
Se manifester	G5	95%
Mettre les deux mains haut-bas	G6	82%
S'accroupir	G7	80%
Tirer avec un pistolet	G8	90%
Jeter un objet	G9	75%
Changer d'arme	G10	92%
Coup de pied	G11	80%
Mettre lunettes de vision nocturne	G12	96%

3.5.3 Reconnaissance gestuelle

Pour vérifier la robustesse de la machine d'apprentissage de gestes, nous avons décrit verbalement les gestes entrainés pour 10 personnes. Ensuite, nous avons mesuré le taux de reconnaissance de la machine lorsque les individus ont exécuté chaque geste 10 fois. Des résultats excellents ont été obtenus dans la majorité des gestes, notons que même pour les gestes les plus délicats on a obtenu des résultats satisfaisants, comme indiqué dans le Tableau 3.2. En particulier, les gestes avec contrainte de temps présentent les plus grandes variations de performance, bien que les taux de reconnaissance pour les gestes avec contrainte de temps G05 et G12 sont au-dessus de 80%. Nous observons aussi que notre méthode permet d'obtenir de meilleurs taux de reconnaissance des gestes quand ces derniers sont effectués plus lentement.

3.5.4 Performance

Pendant la phase de prétraitement, le seule, bien que petit, blocage est le calcul des fonctions classifieurs binaires SVM. Pour un grand ensemble d'apprentissage, avec près de 8000 exemples de poses clés de 20 classes, les 20 fonctions ont été calculées en 13 secondes. Notons que ces fonctions ne doivent être calculées qu'une fois, si l'ensemble d'apprentissage reste inchangé.

Au cours des phases de d'apprentissage et de reconnaissance, l'exécution a permis une interaction en temps réel: la machine d'apprentissage de poses clés, composée de plusieurs classifieurs binaires SVM, était capable de reconnaître facilement les poses clés à 30 tps (le taux de trame maximum du capteur Kinect) sur un ordinateur portable Core I5 à 2,53 GHz.

Aussi, d'une part, la plupart des gestes sont composés de quelques poses clés, générant des arbres avec de très faibles profondeurs. D'autre part, chaque largeur d'arbre dépend du nombre de gestes entrainés, ce qui est également un nombre faible dans la plupart des cas. La formulation de la forêt de décision conduit à une complexité de recherche très faible dans la pratique et n'a pas d'incidence sur le temps total d'exécution.

3.5.5 Limites

La plupart des problèmes de robustesse étaient principalement dues à deux raisons, à savoir, le suivi de squelettes et les petits mouvements.

Avec l'utilisation d'une seule Kinect, la personne doit être en face du capteur, puisque les positions latérales peuvent occlurent les articulations, ce qui dégrade la qualité du squelette. En outre, le traqueur de squelettes peut générer différents squelettes pour différentes personnes effectuant la même pose. Ces différences peuvent dégrader l'invariance des descripteurs de pose. Par opposition au graphe d'action [26], notre méthode est limitée à des gestes composés de poses clés distinctives. Les gestes qui exigent des mouvements subtils peuvent être gênant pour nos machines d'apprentissage.

3.6 Conclusion

Dans ce chapitre, nous avons présenté une méthode pour la reconnaissance des poses en temps réel à partir d'un flux de squelettes extrait du capteur de profondeur Kinect. Chaque pose est décrite en utilisant une représentation angulaire des articulations du squelette. Ces descripteurs ont été utilisés pour identifier des poses clés à travers un classifieur SVM multi-classes**.** Ces poses clés sont utilisées pour reconnaitre les gestes à travers une forêt de décision.

Au cours de la phase de reconnaissance de nos tests, l'exécution a permis une interaction en temps réel. La machine d'apprentissage de poses clés a pu facilement reconnaître les poses clés à 30 tps.

Des résultats très satisfaisants ont été obtenus dans la majorité des gestes. Notons que, même pour les gestes les plus délicats, nous avons obtenu des résultats satisfaisants. L'approche utilisée a permis l'interaction en temps réel.

Conclusion générale

Dans ce travail, on a proposé un algorithme de détection de personnes à partir des images de profondeur de la Kinect, le capteur utilisé dans tous nos tests, et un autre algorithme d'identification de postures et de mouvements à partir du flux de squelette fournis par le capteur.

L'algorithme de détection de personnes à partir des images de profondeur utilisé est appelé HOD (histogramme des profondeurs orientées) inspiré du célèbre détecteur HOG (histogramme des gradients orientés) pour les images RGB.

Le HOD, qui représente l'histogramme des profondeurs orientées, code localement la direction des changements de profondeur et repose sur une recherche multi-échelle de profondeur informée qui conduit à une accélération de 3 fois le processus de détection. Le résultat est un détecteur de personnes qui réalise un taux de précision de 85% dans une plage de presque deux fois plus grande que celle spécifiée par le constructeur du capteur. Nous avons en outre mené des expériences comparatives d'analyser la contribution des données de profondeur par rapport aux méthodes purement visuelles. Le HOD implémenté sur un ordinateur portable Core I5 à 2,53 GHz surpasse toutes les autres approches de détection et il est exécuté à 10 tps.

Pour le problème d'identification de postures et de gestes, on a utilisé un descripteur de pose qui représente de façon concise une pose du corps humain où chacune est décrite en utilisant une représentation angulaire des articulations du squelette. Ces descripteurs servent à identifier des poses clés à travers un classifieur SVM multi-classe**.** Par la suite les poses clés ont été utilisées pour reconnaitre les gestes à travers une forêt de décision.

Au cours des phases de l'entraînement et de la reconnaissance de nos expériences, l'exécution a permis une interaction en temps réel: la machine d'apprentissage de poses clés, composé de plusieurs classifieurs binaires SVM, était facilement capable de reconnaître les poses clés à 30 tps (le taux de trame maximum du capteur Kinect).Des résultats excellents ont été

obtenus dans la majorité des gestes, notons que même pour les gestes les plus délicats on a obtenu des résultats satisfaisants

Aussi, la formulation de la forêt de décision conduit à une complexité de recherche très faible dans la pratique et n'a pas d'incidence sur le temps total d'exécution.

Ce travail nous a permis de s'ouvrir sur un certain nombre de perspectives :

Le premier point qui concerne la détection de personnes, combiner l'image RGB riche avec l'information couleur et la texture avec l'image de profondeur peut générer un descripteur plus robuste, aussi implémenter l'algorithme sur un GPU peut accélérer la recherche par la suite diminuer le temps d'exécution.

Le deuxième point qui concerne l'identification de postures et de gestes et comme les algorithmes d'extraction et de suivi de squelette encore font face à une grande quantité de bruit du capteur, la robustesse est une question principale pour les travaux futurs. Un problème commun à ces algorithmes est l'occlusion 3D de certaines articulations, obligeant l'utilisateur d'être en face à la caméra pour éviter d'avoir des squelettes aberrants. Travailler avec deux ou plusieurs capteurs Kinect pourrait être utile pour capturer et traiter le flux de squelette d'une façon robuste.

L'algorithme de reconnaissance de gestes présenté ici peut être amélioré dans deux directions différentes. Tout d'abord, l'utilisation du temps dans la reconnaissance des gestes peut être améliorée pour distinguer des gestes complexes, utilisant des machines SVM complémentaires, et peut-être en ajoutant la vitesse des articulations. Deuxièmement, la génération automatique des poses clés à partir des gestes non reconnus peut grandement faciliter l'usage de l'interface. Dans ce cadre, l'ordinateur doit être en mesure de choisir la meilleure série de poses clés pour l'entraînement, afin d'obtenir de bons résultats dans la reconnaissance des gestes.

Références bibliographies

[1] Corradini, A. et Cohen, P. Multimodal speech-gesture interface for handfree painting on a virtual paper using partial recurrent neural networks as gesture recognizer. In Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on, volume 3, pages 22932298. IEEE, 2002.

[2] http://www.larousse.fr/encyclopedie/rechercher?q=geste

[3] Yang, H.-D., Park, A.-Y. et Lee, S.-W. Robust Spotting of Key Gestures from Whole Body Motion Sequence. In Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition, FGR '06, pages 231236. IEEE Computer Society, 2006.

[4] Mcneill, D. Hand and mind. What gestures reveal about thought. University Of Chicago Press, August 1992.

[5] Pavlovic, V. I., Sharma, R. et Huang, T. S. Visual Interpretation of Hand Gestures for Human-Computer Interaction. A Review. IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, n7, pages 677695, 1997.

[6] Shotton, J., Fitzgibbon, A. W., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A., 2011. Real-time human pose recognition in parts from single depth images. In: CVPR. pp. 1297-1304.

[7] J. B. MacQueen. "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press,no 1, pp281-297.1967.

[8] Berrani, S.-A., Amsaleg, L., & Gros, P. Recherche par similarités dans les bases de données multidimensionnelles : panorama des techniques d'indexation.

[9] Vladimir N. Vapnik. Statistical Learning Theory. Wiley-Interscience, 1998.

[10] D.Rumelhart and J.McClellan, Parallel Distributed Processing Explorations in the Microstructure of Congntion. MIt Press.

[11] Quinlan, J. R. Induction of decision trees. Machine Learning, 1986.

[12] L. Breiman. Random forests. Machine Learning, 45(1):5–32, 2001.

[13] L. Spinello and K. Arras, "People Detection in RGB-D Data," in Proceedings of IROS2011, pp. 3838–3843. 2011.

[14] F. C. Crow, \Summed-area tables for texture mapping," SIG-GRAPH Comput. Graph. vol. 18, pp. 207 12, January 1984.

[15] Yujie, Shen Zhonghua Hao, Pengfei Wang, Shiwei Ma and Wanquan Liu. A Novel human detection approach based on depth map via Kinect**,** 2013.

[16] Kirsch, R. (1971). *"Computer determination of the constituent structure of biological images"*. Computers and Biomedical Research 4: 315–328.

[17] T. Ojala, M. Pietikäinen, and D. Harwood (1994), "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions", Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR 1994), vol. 1, pp. 582 - 585.

[18] S. Ikemura, and H. Fujiyoshi. Real-Time Human Detection using Relational Depth

Similarity Features, ACCV. pp. 25-38, 2010.

[19] Yoav Freund et Robert Schapire, « A desicion-theoretic generalization of on-line learning and an application to boosting », *Journal of Computer and System Sciences*, vol. 55, nº 1, 1997, p. 119-139

[20] Lu Xia, Chia-Chih Chen, and Aggarwal J.K. Human detection using depth information by Kinect. Computer Vision and Pattern Recognition Workshops, pp.15-22, 2011.

[21] Canny, J., *A Computational Approach To Edge Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8:679-714 (1986).

[22] Ferda Ofli et al. Sequence of the most informative joints (smij) : A new representation for human skeletal action recognition. Journal of Visual Communication and Image Representation, 2013.

[23] Mohamed E. Hussein et al. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. International Joint Conference on Artificial Intelligence, 2013.

[24] Svetlana Lazebnik et al. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. IEEE CVPR, 2, 2006.

[25] Wanqing Li et al. Action recognition based on a bag of 3d points. Int'l workshop on CVPR (CVPR4HB), 2010.

[26] Wanqing Li et al. Expandable data-driven graphical modeling of human actions based on salient postures. IEEE Transactions On Circuits And Systems For Video Tech, 18, 2008.

[27] Jiang Wang et al. Mining actionlet ensemble for action recognition with depth cameras. IEEE Conference on Computer Vision and Pattern Recognition, pages 1290–1297, 2012.

[28] Omar Oreifej and Zicheng Liu. Histogram of oriented 4d normals for activity recognition from depth sequences. IEEE ICPR, 2013.

[29] Sylvain Calinon and Aude Billard. Stochastic gesture production and recognition model for a humanoid robot. IROS IEEE, 2004.

[30] Clément Réverdy. Modèles de Markov Cachés (HMM) pour de la reconnaissance de gestes humains. Machine Learning. 2014.

[31] Aditya Ramamoorthy et al. Recognition of dynamic hand gestures. 2003.

[32] Ferdinando Samaria and Steve Young. Hmm-based architecture for face identification. Image and Vision Computing, 1994.

[33] J. Yamato et al. Recognizing human action in time-sequential images using hidden markov model. 1992.

[34] Richard Bowden and David Windridge. A linguistic feature vector for the visual in terpretation of sign language. 2004.

[35] Mitra and Tinku Acharya. Gesture recognition : A survey. IEEE Transactions On Systems,Man, And Cybernetics, 2007.

[36] Faisal Bashir et al. Hmm-based motion recognition system using segmented pca. ICIP IEEE, 2005.

[37] *Pearson, K.* (1901). *"On Lines and Planes of Closest Fit to Systems of Points in Space"* Philosophical Magazine 2 (11): 559–572.

[38] Frédéric Bevilacqua et al. Continuous real time gesture following and recognition. Kopp S and Wachsmuth I. editors ”Gesture in Embodied Communication and Human-Computer Interaction”, 5934 :73–84, 2010.

[39] Samsu Sempena et al. Human action recognition using dynamic time warping. International Conference on Electrical Engineering and Informatics, 2011.

[40] Shah Muhammed Abid Hussain and A. B. M. Harun ur Rashid. User independent hand gesture recognition by accelerated dtw. ICIEV, 2012.

[41] Pierre-François Marteau et al. Sous échantillonnage et machine à noyaux élastiques pour la classification de données de mouvement capturé. EGC, 2014.

[42] Miranda, L., Vieira, T., Martinez, D., Lewiner, T., Vieira, A. W., Campos, M. F. M., 2012. Real-time gesture recognition from depth data through key poses learning and decision forests. In: Sibgrapi. IEEE, Ouro Preto, MG, pp. 268-275.

[43] Raptis, M., Kirovski, D., Hoppe, H., 2011. Real-time classification of dance gestures from skeleton animation. In: SCA. pp.147-156.

[44] Dalal, N.; and Triggs, B. Histograms of oriented gradients for human detection. Computer Vision and Pattern Recognition, pp.886-893, 2005.

[45] https://msdn.microsoft.com/en-us/library/hh855347.aspx

[46] K. Konolige and P. Mihelich. Technical description of kinect calibration. http://www.ros.org/wiki/kinect_calibration/technical

[47] http://opencv.org/

[48] http://openni.ru/openni-sdk/

[49] http://www.informatik.uni-freiburg.de/~spinello/sw/rgbd_people_unihall.tar.gz

[50] Vapnik, V. The Nature of Statistical Learning Theory. Springer, 2000.